Non-Intrusive Measurement in Packet Networks and its Applications

Ming, Leung Chi

Non-Intrusive Measurement in Packet Networks and its Applications

By

Leung Chi Ming

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Department of Electronic Engineering

Queen Mary, University of London

February 2004

ABSTRACT

Network measurement is becoming increasingly important as a means to assess the performance of packet networks. Network performance can involve different aspects such as availability, link failure detection etc, but in this thesis, we will focus on Quality of Service (QoS). Among the metrics used to define QoS, we are particularly interested in end-to-end delay performance. Recently, the adoption of Service Level Agreements (SLA) between network operators and their customers has become a major driving force behind QoS measurement: measurement is necessary to produce evidence of fulfilment of the requirements specified in the SLA.

Many attempts to do QoS based packet level measurement have been based on Active Measurement, in which the properties of the end-to-end path are tested by adding testing packets generated from the sending end. The main drawback of active probing is its intrusive nature which causes extra-burden on the network, and has been shown to distort the measured condition of the network. The other category of network measurement is known as Passive Measurement. In contrast to Active Measurement, there are no testing packets injected into the network, therefore no intrusion is caused. The proposed applications using Passive Measurement are currently quite limited. But Passive Measurement may offer the potential for an entirely different perspective compared with Active Measurements

In this thesis, the objective is to develop a measurement methodology for the end-to-end delay performance based on Passive Measurement. We assume that the nodes in a network domain are accessible. For example, a network domain operated by a single network operator. The novel idea is to estimate the local per-hop delay distribution based on a hybrid approach (model and measurement-based). With this approach, the storage measurement data requirement can be greatly alleviated and the overhead put in each local node can be minimized, so maintaining the fast switching operation in a local switcher or router.

Per-hop delay distributions have been widely used to infer QoS at a single local node. However, the end-to-end delay distribution is more appropriate when quantifying delays across an end-to-end path. Our approach is to capture every local node's delay distribution, and then the end-to-end delay distribution can be obtained by convolving the estimated delay distributions. In this thesis, our algorithm is examined by comparing the proximity of the actual end-to-end delay distribution with the estimated one obtained by our measurement method under various conditions. e.g. in the presence of Markovian or Power-law traffic. Furthermore, the comparison between Active Measurement and our scheme is also studied.

Network operators may find our scheme useful when measuring the end-to-end delay performance. As stated earlier, our scheme has no intrusive effect. Furthermore, the measurement result in the local node can be re-usable to deduce other paths' end-to-end delay behaviour as long as this local node is included in the path. Thus our scheme is more scalable compared with active probing.

ACKNOWLEDGEMENTS

I would like to take the opportunity to thank my supervisor Dr. John Schormans who has given me intensive supervision through my study, thanks for his patience and time for his guidance, advice and suggestions and critical reading without which there would be no thesis.

I also would like to thank everybody in the department of Electronic Engineering at Queen Mary, University of London for creating a nice and friendly working environment, especially thanks to Athen Ma, Siew Bee Yeap, Choo Chiap Chiau, Girum Aklilu, Nalinda Somasiri, Wee Kian Toh and Yasir Alfadhl for their help and friendship.

A great gratitude is dedicated to my family, especially my parents, my sister and brothers, and all my nieces and nephews. I am grateful to them for the continuous encouragement and support.

The last but not least, my special thank-you to my dearest wife, Joanna who helps me through the difficult times and gives me her unconditional love and support all the time.

Leung Chi Ming

UK

February, 2004

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLES

GLOSSARY

ATM	Asynchronous Transfer Mode
BMAP	Batch Markovian Arrival Process
BW	Bandwidth Measurement Scheme
EM	Expectation Maximization
ER	Excess-Rate
GPS	Global Position System
ICMP	Internet Control Message Protocol
IEFT	Internet Engineering Force
IPPM	IP Performance Metric
LDT	Large Deviation Theory
LRD	Long Range Dependent
MBAC	Measurement-Based Admission Control
MIB	Management Information Base
MMPP	Markov Modulated Poisson Process
MPLS	Multiprotocol Label Switched Routing
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
MVT	Maximum Variance Theory
NMS	Network Management System
NOC	Network Operating Centre
NTP	Network Time Protocol
OWAMP	One-Way Active Measurement Protocol
QL	Queue Length Measurement Scheme
QoS	Quality of Service
RMON	Remote Monitoring
RTT	Round-Trip Time
SLA	Service Level Agreements
SNMP	Simple Network Management Protocol
SRD	Short Range Dependent
TTL	Time-To-Live
WFQ	Weighted Fair Queuing
WRR	Weighted Round Robin

LIST OF MATHEMATICAL SYMBOLS

$F(.)$	probability density function;
	WFQ's virtual finish time
C	the service rate of the buffer
ϕ_i	the scheduler's weight for queue i
D	end-to-end delay
	the probability a flow is delayed
$Q(.)$	queue length distribution
$P(.)$	probability function
$\Lambda(.)$	moment generating function
$\psi(.)$	Gaussian tail function
κt	mean of the increment in Maximum Variance Theory
v_t	variance of the increment in Maximum Variance Theory
$R_{fb}(.)$	autocorrelation function for Fractional Brownian Motion
H	Hurst parameter
c_{pm}, c_{bm}	packet-scale decay constant, burst-scale decay constant (Markovian traffic)
c_{bp}	burst-scale decay constant (Power-law traffic)
q_{high}, q_{low}	measurement data in QL measurement data
$freq_{high}$, $freq_{low}$	the number of occurrence of hitting $queue_{high}$ and $queue_{low}$ region
B	the largest buffer size during the measurement
η_{bm}, η_{bp}	burst-scale decay rate (Markovian traffic), (Power-law traffic)
e	packet size
e_v	average packet size
x	queue length
$Q_{end\text{-}to\text{-}end}(.)$	end-to-end delay distribution
$R(.)$	WFQ's round number
δ_i	the predefined threshold in BW Measurement
q_p	partition point in QL Measurement
N	the number of source
R	maximum transmission rate of a ON/OFF source
R_{on}	mean rate in the ON state (aggregate model)
s	the probability of being silent for another time unit
α	ON/OFF source's activity factor
ρ	buffer utilization
$T_{(on)}$	mean ON time (aggregate model)

$T_{(off)}$	mean OFF time (aggregate model)
F	the rate of flow arrivals
A_p	mean load in packets/s
R_{off}	mean rate in the OFF state (aggregate model)
A	the offered traffic in Erlangs
x_{pb}	knee point between packet and burst-scale queuing region
C'	new bandwidth
R_{on}'	new mean rate in the ON state (aggregate model)
R_{off}'	new mean rate in the OFF state (aggregate model)
s'	new the probability of being silent for another time unit
a'	new probability of generating another excess-rate arrival in the ON state
a	the probability of generating another excess-rate arrival in the ON state
D'	new probability a flow is delayed
η_b'	new burst-scale decay rate
c_b'	new burst-scale decay constant
N_o	minimum number of active sources for burst-scale queuing

Chapter 1 Introduction

1.1. Motivation behind This Research

Network measurement is essential for a variety of objectives: network planning, performance evaluation etc. This ensures that service/network providers provide satisfactory QoS. Various performance metrics have been defined to assess the QoS, e.g. packet loss, jitter, latency. Among these performance metrics, the issue of the measurement for delay performance is our main interest. We propose a novel Passive Measurement scheme for the end-to-end latency measurement. This scheme is not exclusive to delay measurement, but can also be found useful in an application to bandwidth adjustment.

Currently, network measurement can be classified into two basic categories: *Active Measurement* or *Passive Measurement*. Broadly speaking, in Active Measurement, the path properties are tested by introducing testing packets into the network. Owing to these extra testing packets, Active Measurement is well known for its inherent unfavourable side-effect: an *intrusive* nature. Clearly, the extra load introduced into the network either causes extra burden or distorts the original condition. In contrast to Active Measurement, no testing packets are required in Passive Measurement, therefore there is no intrusive effect introduced into the network. Certainly, each measurement scheme has its own merits and shortcomings. One may find Active Measurement more suitable or easier to implement in end-to-end performance testing. This is reflected by the fact that the current end-to-end performance measurement is mainly based on it. Whilst Active Measurements may provide good estimates of some metrics e.g. average network delay, with other applications, they may require a large amount of probe traffic to obtain useful information e.g. in the construction of statistical models, in the evaluation of SLAs etc. The intrusive effect may then be substantial [TIM03]. This is unwelcome by most network administrators, and the network nodes may be configured in such a way that the probing packets receive different treatment. As a consequence, this results in an untrue measurement result.

Passive Measurement, or *non-intrusive measurement*, on the other hand, does not have any intrusive effect. The main application of Passive Measurement can be found on the traffic volume measurement. Although passive delay measurement, in particular at network nodes, is not widely available, *it offers the potential for an entirely different perspective* [BAR02]. QoS assessment has long been emphasized, but there is still no standard to govern how the measurement should be

carried out. This motivates this research work to study the employment of Passive Measurement for QoS measurement. We have developed a measurement methodology for the end-to-end delay measurement and we also explored its application to different aspects.

It is known that routers are always optimised for the relatively simple task of forwarding packets. This means that routers often make bad measurement targets as it is unwise to put much burden on the routers for measurement purposes. This imposes a challenge on designing our measurement scheme. Simplicity, low computational complexity and overheads must always be the first criteria for performing any measurement in the network nodes. In addition, the size of the measurement result is also an issue. The huge amount of measurement data imposes stringent storage requirements on the network nodes and causes congestion to the Network Operating Centre (NOC) during data transfer period. NOC is normally responsible for gathering the measurement results for analysis or producing reports. Therefore, a small size of measurement results is certainly beneficial. Our approach combines the measurement method and queue length distribution model, whereby the queue is passively/non-intrusively monitored at each network node. The measurement result is used to estimate model parameters of the queue length distribution. With this *hybrid* approach, it alleviates the measurement burden on the network nodes and ensures a small volume of measurement data.

1.2. Objective of This Thesis

The objectives of this research work are threefold:

- The main objective is to develop and analyse our novel measurement methodology for the delay performance evaluation. Our measurement methodology is termed as Queue Length Measurement, or *QL* Measurement in short term. The effectiveness of our measurement scheme is examined through extensive simulations in various scenarios.
- The delay distribution is an effective means for delay performance assessment, like SLA's end-to-end delay performance validation. We justify our measurement scheme according to its ability in capturing the delay distribution.
- The application of our measurement methodology in capacity planning/bandwidth adjustment. Sufficient bandwidth allocation is a basic requirement to guarantee QoS. We further apply our measurement scheme to determine the effect of bandwidth adjustment.

1.3. Novelty and Contribution of This Research

In this research work, we developed a novel Passive Measurement by an approach of queue length monitoring. This approach provides a simple measurement scheme with small requirement for data storage. Due to its non-intrusive nature, our measurement methodology does not possess the unfavourable intrusive properties found in Active Measurement. Our novel measurement methodology "QL Measurement" is to capture the local queue length distribution. The local queue length distributions provide useful information in different areas:

- Local QoS inference and identify/locate the problem in the network domain. The detail can be found in Chapter 4
- Producing the end-to-end delay distribution. The distribution can be useful for Service Level Agreement (SLA) validation and delay performance assessment as discussed in Chapter 5
- Traffic model parameter estimation. This provides information to analyse the effect of changing bandwidth as shown in Chapter 8.

1.4. Layout of This Thesis

This thesis comprises of four parts: background of the research, proposed method, evaluation and the applications. It is organised as follows:

Chapter 2 revisits the motivation of the network measurement in more details. Network measurement is an essential process found in the normal network operations. We review the network management and its tools: SNMP and MIB. We also review two basic measurement schemes: Active and Passive Measurement and their variants found in the industry. Their features, advantages and shortcomings will also be highlighted and compared. The review of the measurement-based research topics is finally presented.

Chapter 3 accounts for the network delay components and their behaviour. The latency measurement methods are highlighted and their properties are discussed. There are shortcomings in the current latency measurement methods and this motivates us to develop our measurement scheme QL Measurement. QL Measurement is performed by monitoring the queue length. The closely-related works based on queue length monitoring will be reviewed.

Chapter 4 presents QL Measurement for the local queue length distribution estimation. The queue length distribution model is crucial to our scheme. Both queuing discipline and traffic characteristics have great impact on the queue length distribution. Prior to the introduction of our QL Measurement scheme, we have a review of two major classes of traffic model, i.e. Short-Range Dependent SRD traffic and Long-Range Dependent LRD traffic. SRD traffic, like Markovian traffic model, has been long used to model voice traffic. However, recent experimental result demonstrates that Markovian traffic model fails to represent data traffic which has Long-Range Dependent properties. Both traffic models lead to different shapes of the queue length distribution. We illustrate our algorithm to capture the queue length distributions based on QL measurement result and how to use the queue length distribution to estimate the end-to-end latency performance.

Chapter 5 presents the evaluation results of our QL Measurement scheme. The effectiveness of the scheme is examined for different aspects and conditions through a series of experiments.

Chapter 6, in this chapter, we consider a heterogeneous network carrying both Markovian and Power-law traffic. The traffic classes are assumed to be separated by a scheduler. Due to the presence of the scheduler, the phenomenon of the bandwidth stealing inevitably occurs. This may affect the accuracy of our end-to-end latency estimation mechanism. To improve our measurement mechanism, we introduce a measurement mechanism called "Bandwidth Measurement" or "BW Measurement" for brevity, to estimate the actual bandwidth received by a traffic class. This information is useful to determine the end-to-end delay distribution as discussed in Chapter 5. Again, simulation studies are presented to validate our idea.

Chapter 7 presents the comparison between active probing with our measurement scheme on two aspects: the performance of capturing the end-to-end delay distribution and SLA's delay performance validation. The end-to-end network performance currently relies on active probing. As far as we aware, there is not much work performed to compare the Active Measurement and Passive Measurement. In this chapter, we will study the comparisons of both schemes through simulations.

Chapter 8 presents the further possible application of our non-intrusive measurement scheme. Apart from end-to-end delay distribution measurement, we apply our measurement on the bandwidth adjustment problem. The methodology exploits the well-developed closed-form analytical model (Excess Rate Analysis). We introduce the factor "knee point" between packet-scale queuing and burst-scale queuing to this analytical model to enhance the accuracy. This is justified with the simulation results. With the use of this enhanced formula, the ON/OFF traffic characteristics can be

estimated by the measurement. Then, the effect of changing bandwidth on the queuing behaviour can be estimated by re-applying this enhanced formula. Again, this novel idea is tested with simulations.

Chapter 9, in this final chapter, the discussions and summary of this work are given. In addition, some of the possible ways to continue this research are also addressed.

Chapter 2 Telecommunication Network Measurement

This chapter discusses network measurement in different aspects: motivations, usage and methods.

2.1 Why Network Measurement?

As mentioned in Chapter 1, the reason behind performing network measurement can be found as follows:

- Capacity planning
- Assisting network administrators to provide information for the network operation centre and engineering with data that can not be found from the network management system
- Value-added service – SLA validation – providing reports to customers
- Performance monitoring – fault detection
- Usage-based billing

Generally speaking, measurement is performed for ensuring network performance. Without measurements, there will be no objective record or benchmark of how a network behaves. Measurements show whether changes improve or degrade the network's performance, and by how much [BRO01]. When service performance degradation occurs, there are common questions asked by the network administrator or user [FRO99] such as:

- Why is the application so slow?
- Where do the delays come from?
- What can be done to improve performance?
- Will additional bandwidth help?

Information must be collected by measurements in order to provide answers to these questions. For example, measurement for capacity planning provides the information for network operators to calculate necessary capacity for required workload, in order to avoid congestion in the network. The degradation of performance may cause long packet delay time, jitter or larger packet loss ratio. For packet voice application, the voice quality is not acceptable if the packet delay time is larger than 300m sec [IEC03]. The application performance degradation can be caused by numerous factors such as insufficient bandwidth, link failure etc. Network measurement is necessary to monitor the network performance, analyse and identify the faults.

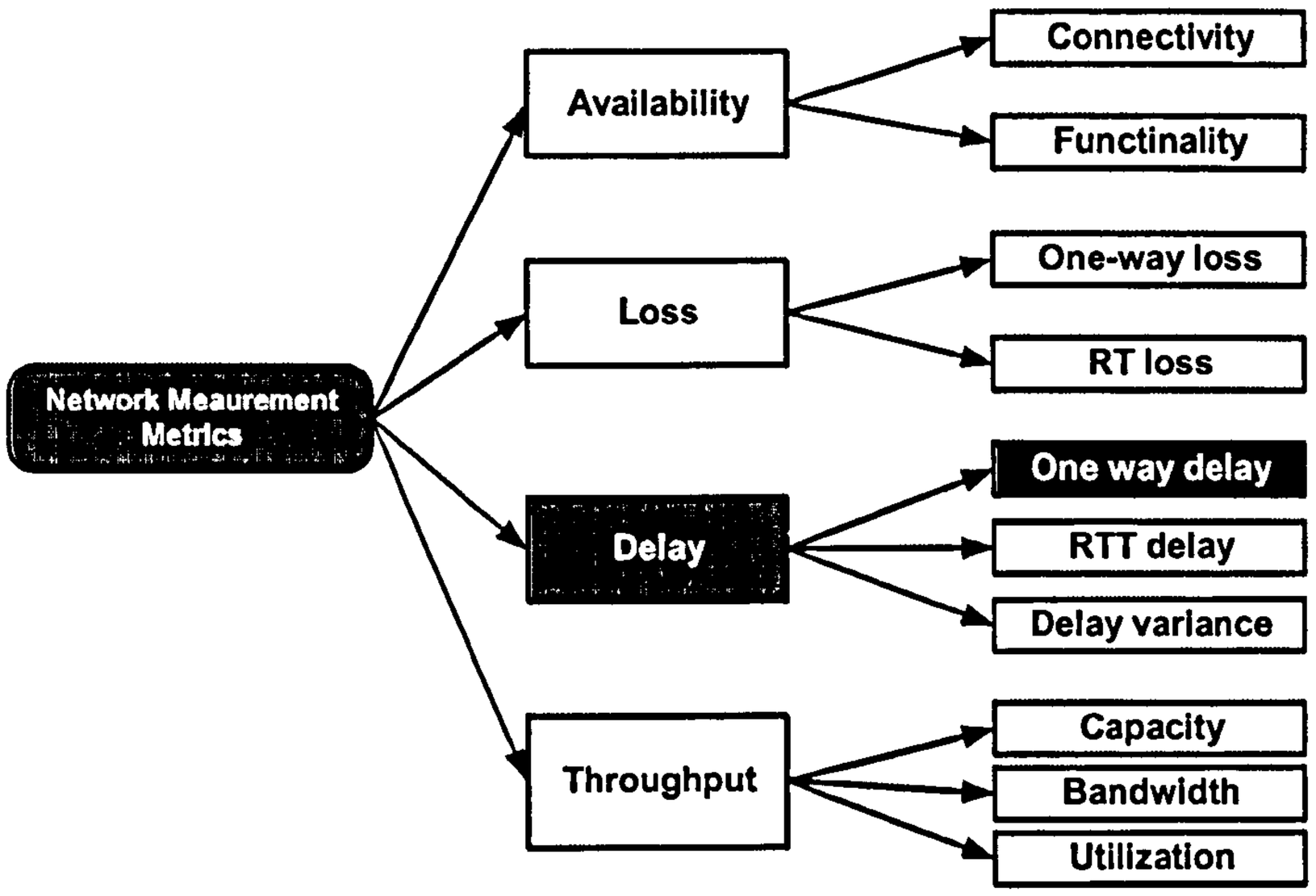


Figure 2-1 Network measurement metrics [HON03]

Figure 2-1 depicts the common network measurement metrics. These factors are availability, packet loss, delay and throughput. Network measurement may be performed in order to assess all of them. With reference to Figure 2-1, the delay metrics can be end-to-end (one-way) delay, Round trip delay and delay jitter. Among these delay performance metrics, in this thesis, we are particularly interested in the end-to-end (one-way) delay performance measurement.

Apart from network performance assessment, network measurement provides necessary information for usage-based billing and SLA validation.

2.1.1 Service Level Agreement

A SLA specifies the expected service performance that a customer expects to receive. Normally, a customer can be an organization or another service provider. The SLA aims to increase the level of competitiveness of the service provider. The common items found in this agreement can be Mean Time Between Failure (MTBF) i.e. the average time between two consecutive failures to happen, Mean Time To Repair (MTTR), i.e. the average time required to fix the problem or failure, Availability, i.e. the likelihood of the service is available to the customer and QoS performance parameter, (for example, percentile of the packet loss exceeding a certain limit, percentile of the

packet delay exceeding a specific limit and the delay jitter). Failure to meet the requirement listed in a SLA will incur the remedy to customers [TEL01].

Due to the existence of the SLA, both the service provider and the customer will want to know if the SLA targets are being met. The customer is paying to receive the service levels specified in the SLA, and wants to verify if they are actually getting those service levels. Likewise, the provider has committed to delivering the service levels specified in the SLA, and wants to monitor if this is actually the case. The provider can also use the verification results as a record and as a form of proof of the actual service levels achieved. This information can be useful in case of any disputes with customers.

To verify the SLA, it is essential to define measures and collect the necessary data. However, SLAs for packet networks are still in a stage of infancy. For ATM, although ATM as a customer service has clearly moved beyond the early adoption market, it has not yet reached maturity [NOR00]. With respect to the QoS metrics in SLA, there is still a lack of measurement tools [PAR01], although several organization bodies are working on setting up standards like IP Performance Metric IPPM Workgroup [PAX98].

Domestic	Transatlantic
05/01 – 51.282ms	05/01 – 85.420ms
04/01 – 54.179ms	04/01 – 81.090ms
03/01 – 53.874ms	03/01 – 90.435ms
02/01 – 52.891ms	02/01 – 87.185ms
01/01 – 54.016ms	01/01 – 81.490ms
12/00 – 52.343ms	12/00 – 83.870ms

Table 2-1 An example of a SLA latency report

Table 2-1 is an example showing how a service provider [WOR01] presents the SLA latency performance to its customer for the period from December 2000 to May 2001. The table lists out the average monthly latency of its domestic US network and the trans-Atlantic connection between New York and London.

From the customer's point of view, this kind of SLA report is unsatisfactory as it fails to provide a clear idea of latency performance that is received. One of the reasons is that there may exist different paths between any end points in the network domain. It is possible that the path used by a particular customer is congested compared with the rest. In addition, the mean latency value is

unable to show the percentile of the packet delay exceeding a pre-defined value that the service provider has guaranteed.

Although network measurement is important, network operators are still having difficulties in employing measurement. These problems will be addressed in Section 2.3 for discussion on the network measurement methods. Measurement targets found in network measurement are quite broad such as link utilization, packet per second, packet loss and latency [PAX98]. *Of these, we restrict ourselves to latency performance measurement.*

2.2 Network Management

Network performance measurement/monitoring comes within the general area of network management. In this section, we present an overview of network management. The International Organization for Standards (ISO) has defined *five key areas* of network management. These are: fault management, configuration management, security management, performance management and accounting management. There are three different management groups to perform the above tasks: 1. Network Provisioning 2. Network Operations and 3. Network Maintenance.

Figure 2-2 depicts a top-down view of network management functions. The role of network operations is to perform the normal daily operations which involve duties such as network measurement/performance management. The functions of network operations are administered by a Network Operations Centre (NOC). NOC is responsible for gathering statistics (e.g. data on traffic, network availability and *network delay*) to generate reports for management, system support, and users, as well as to tune the network for optimum performance.

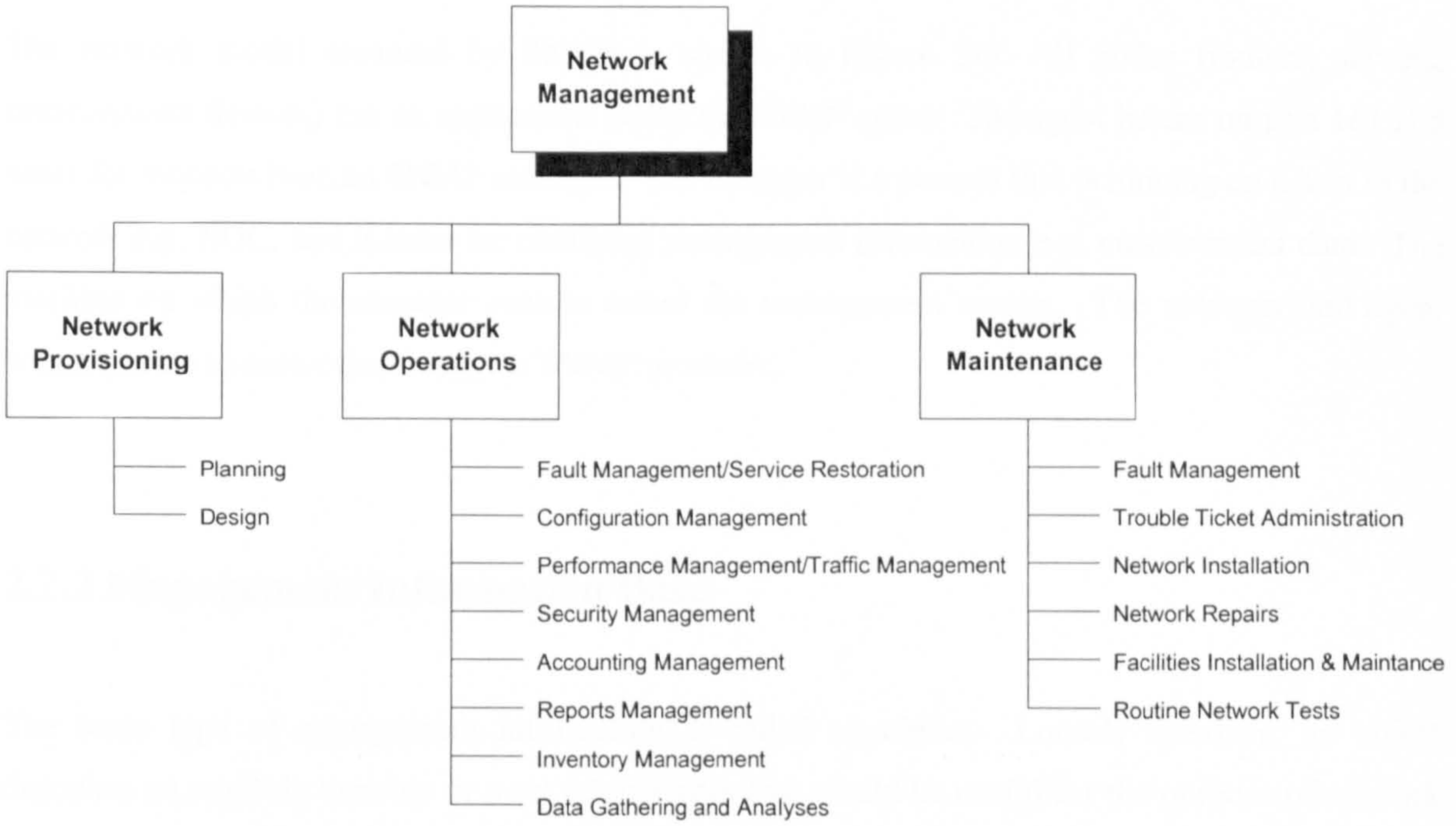


Figure 2-2 Network management functional grouping [SUB01]

To facilitate the network management operations, network management tools were developed. Currently, the prevailing network management tools are based on Simple Network Management Protocol (SNMP), whereas Common Management Information Protocol (CMIP) [SUB01] and Telecommunications Management Network (TMN) are relatively less prevalent.

2.2.1 SNMP

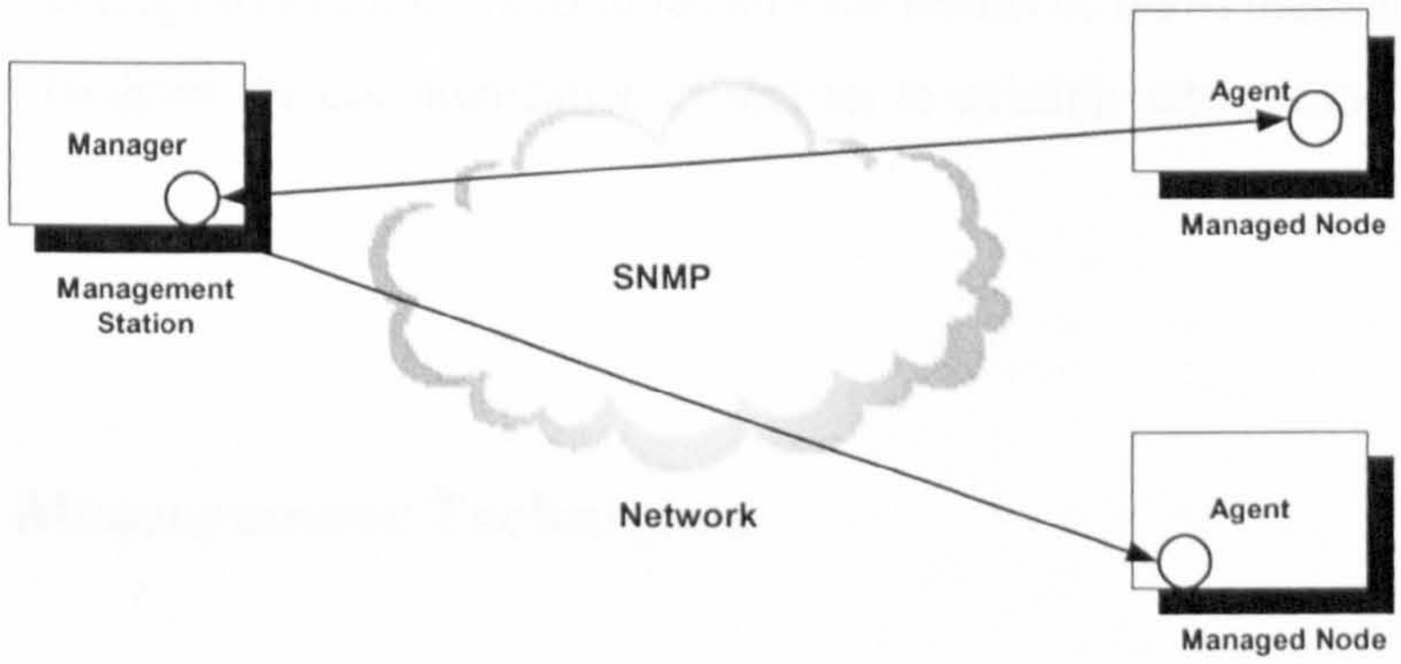


Figure 2-3 SNMP network model

The network model assumed by SNMP is shown in Figure 2-3. All nodes (routers, servers, measurement devices) run an application called the *SNMP agent*. The agent listens on port 161 and waits for requests from an SNMP manager. The manager is a process that is running on a host in the network e.g. NOC, and is used for obtaining management information e.g. measurement data. The machine on which the manager runs is called the management station. The manager and agent communicate to each other using the SNMP protocol.

2.2.2 Management Information Base

The basic type of management information is called an object. Loosely speaking, an object describes an attribute variable or a set of attributes that would be useful for the purpose of network management (e.g. measurement data). These objects are collected into a Management Information Base (MIB). Each type of SNMP agent supports a specific group of MIBs, depending on the types of network elements for which it is providing management information.

MIBs are defined using a language called ASN.1, or Abstract Syntax Notation. ASN.1 defines how to store and transport generic information. The object description for each attribute variable typically consists of a unique name for the attribute, its syntax (whether it is an integer or a string, and so on), and how it is encoded for transporting over the network. The unique name for the object/MIB is determined in a hierarchical manner. Adding new types of items for network management is relatively easy. This can be carried out by defining the new MIB by using the standardized notation. With the object/MIB, the management information on the managed devices can be retrieved/modified by the manager for the management purpose. The IP Performance Metric (IPPM) group is working on defining a MIB to retrieve the results of IPPM metrics, such as one-way delay and loss, to facilitate the communication of metrics to existing network management systems [STE03].

2.3 Network Measurement Techniques

In the previous sections, we briefly explained the role of network management. Network measurement is part of the responsibilities of the network management. We illustrated how the

management information stored in the agents and retrieved by the manager. When referring to measurement, the agent corresponds to the measurement device. This can be the router itself or an additional measurement equipment. The manager will be the NOC which collects the data for analysis. We will not cover the management tools Remote Monitoring (RMON) [SUB01] in which the data may be analysed in the local agent as well.

2.3.1 Active Measurement

Active Measurements are performed by sending testing packets into the network. The testing packets may contain specific fields (e.g. timestamps in Ping) for the measurement purpose. The testing packets act as a stimulus to the network. The network properties/performance is inferred by either measuring the responses to this stimulus from the network (e.g. two-way measurement like Ping) or collecting the result at the remote receiving end (e.g. one-way measurement like One Way Active Measurement Protocol) as shown in Figure 2-4.

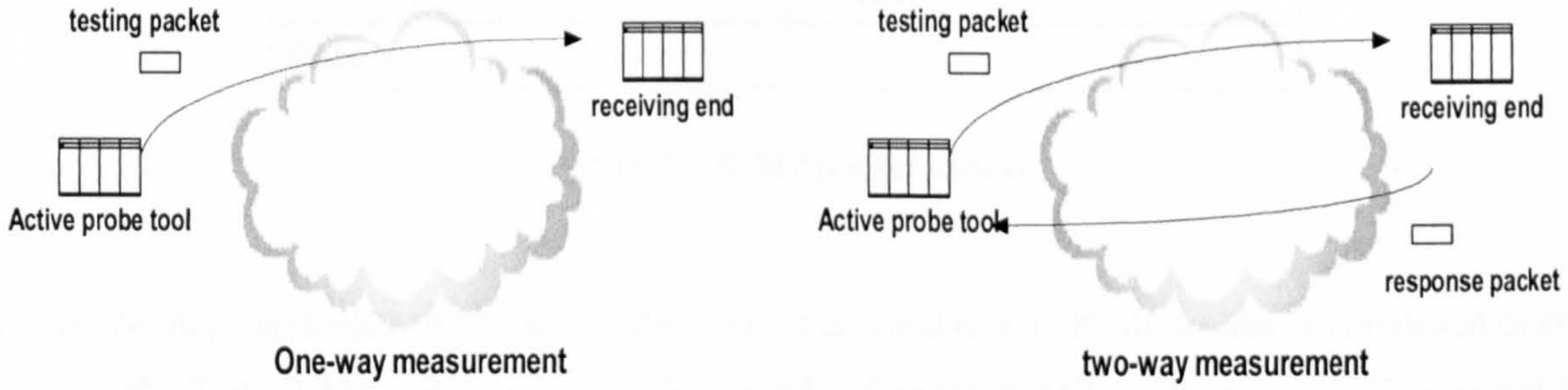


Figure 2-4 One-way and Two-way Active Measurement

Active Measurement are found in different applications such as the measurements for packet delay, packet loss, bottleneck link bandwidth, link bandwidth along the end-to-end path, hop counts or other path characteristics. It is well known that Active Measurement has an intrusive effect on the network. Some measurements may require a large amount of measurement data, therefore more testing packets are required to generate a useful result e.g. SLA validation However, this may add substantial burden to the network [TIM03]. Additionally, the original measuring condition may also be distorted by the extra traffic. This effect is referred to the "*Heisenberg*" effect, having a natural analogy to physics [PAX96a]. The Heisenberg effect makes the measurement error-prone. These two inherent drawbacks of Active Measurement, i.e. potential extra burden to the network and "Heisenberg" effect, restrict itself from being used in large-scale measurements.

Some typical practical Active Measurement tools are highlighted in the following sections.

2.3.1.1 Active Measurement Tools

A. Ping

The most widely used method to investigate network delay is for a measurement device to construct and transmit an Internet Control Message Protocol (ICMP) [SUB01] echo request packet to an echo end. An ICMP header follows the IP header in an IP packet, but it is not considered to be a Layer 4 header like TCP or UDP. Instead, ICMP is considered to be an integral part of IP. Therefore, it is necessary for the vendor to include ICMP during the implementation of IP.

8	16	32 bits
Type	Code	Checksum
Identifier		Sequence number
Data		

Figure 2-5 ICMP packet header

Figure 2-5 depicts the ICMP packet header field. It is noted that an ICMP header is composed of six fields. The Type field together with the Code field define the ICMP packet type. A Ping packet is defined by initialising the Type field to "8" and Code field to "0" (see Appendix A for the whole definition). The ICMP's echo request packet is generated and injected into the network. If the destination receives this Ping packet, it will return an echo packet with Type field = "0" and Code field = "0" to the sending end in accordance with ICMP protocol. The basic applications of Ping are as follows:

- To test the availability of the connection between the source and destination end
- To measure the Round Trip Time RTT

To test the availability of the connection between the source and the destination, the source "*pings*" the destination by sending ICMP's echo request packet. If the destination is not reachable, a Time Out will occur to indicate this event.

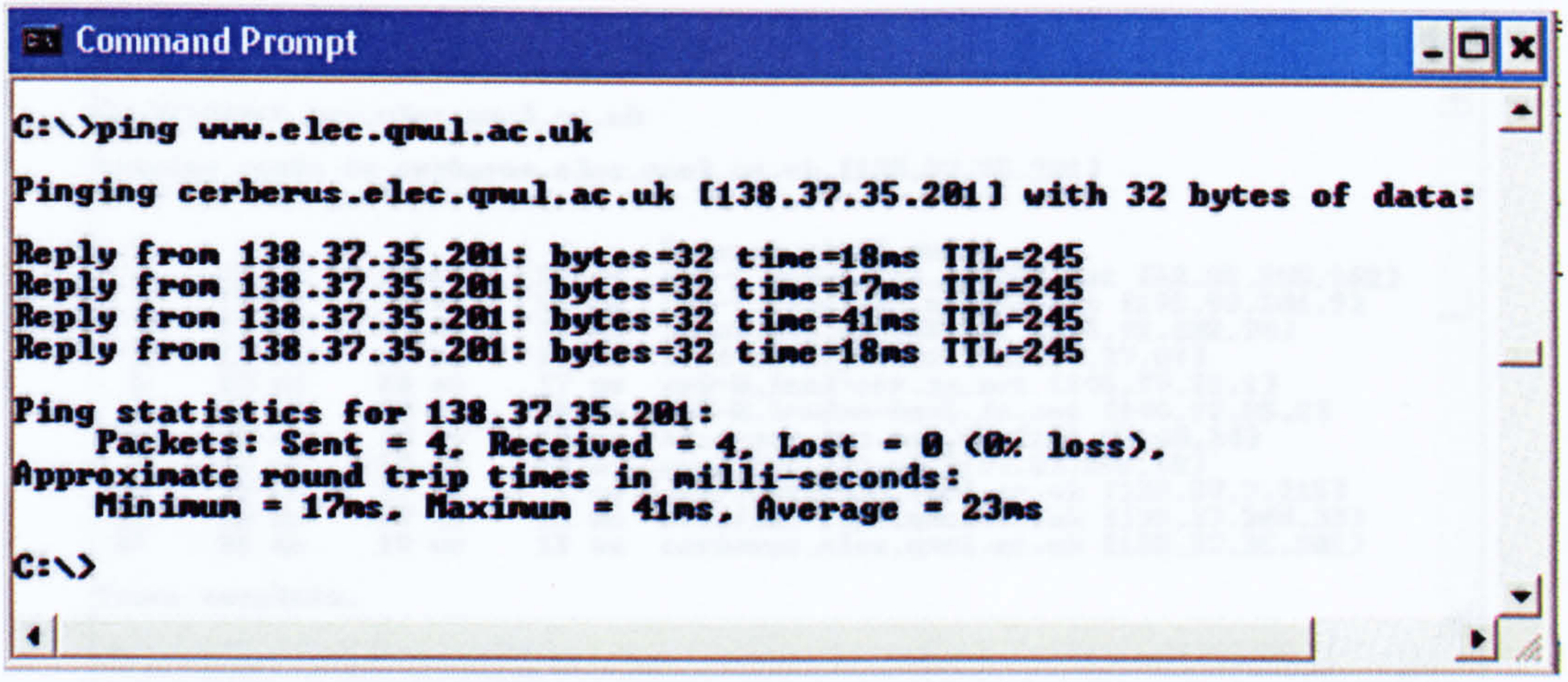


Figure 2-6 A Ping example

Figure 2-6 illustrates a Ping example. The destination was "*pinged*" four times. Each ICMP's echo request packet stores the sending time. After sending out the ICMP's echo request packet, the probing agent will wait for the echo reply packet from the receiving end. The Round Trip Time (RTT) is calculated as the difference between the time the echo request is sent and the time a matching echo reply response is received by the Ping application.

The Time-To-Live (TTL) field indicates the final value of the echo request packet after passing the network. The original TTL value is initialised to 255, the value of TTL is reduced by 1 after passing a node. The TTL field in the ping report can be used to determine the number of hops that the packet passes through. In this case, the packet traverses 10 hops before reaching the destination.

B. Traceroute

The Traceroute technique allows a measurement host to deduce the forward path to the destination. Traceroute makes use of the Time-To-Live (TTL) field in the IP packet (which allows the maximum number of router hops the packet can pass through before being dropped or returned) and ICMP control messages. When the packet is dropped due to the expired TTL, the information from the intermediate node will be returned to the measurement host. A Traceroute example is illustrated in Figure 2-7.

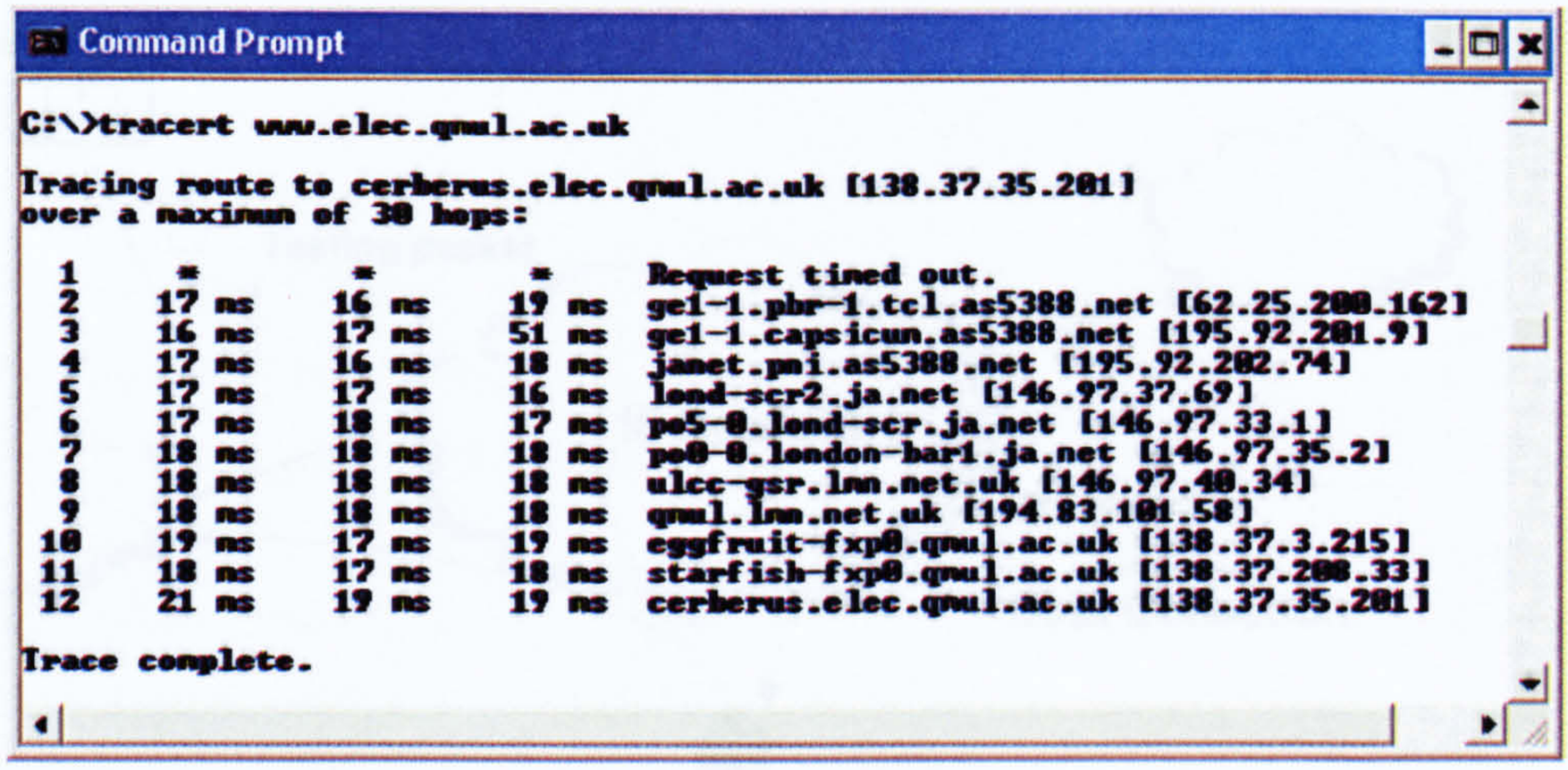


Figure 2-7 A Traceroute example

Initially, the Traceroute application sends out three ICMP's Traceroute packets (with type number = 30) and TTL value = 1. After traversing the first hop, the TTL value is decreased by 1 and so these Traceroute packets will be dropped. The first node sends back the Time Exceeded Message back to the measurement host with the node's information. Therefore, the first node is resolved. Then, another three ICMP's Traceroute packets are sent out, but the TTL value is increased to 2. With the same reason, these three ICMP's Traceroute packets will be dropped in the second hop and the measurement host will obtain the second node's information based on the Time Exceeded Message from this node. The TTL value of the Traceroute packets are gradually increased until the path is identified.

C. One-Way Active Measurement Protocol

Recently, the IETF's IPPM group proposed a new measurement protocol called One-way Active Measurement Protocol (OWAMP) [SHA03]. The measuring tools (probe) are intentionally placed at two measuring points: sending and receiving ends. The probe at the sending end generates and injects the testing packets into the network whereas the probe at the receiving end collects these packets for analysis.

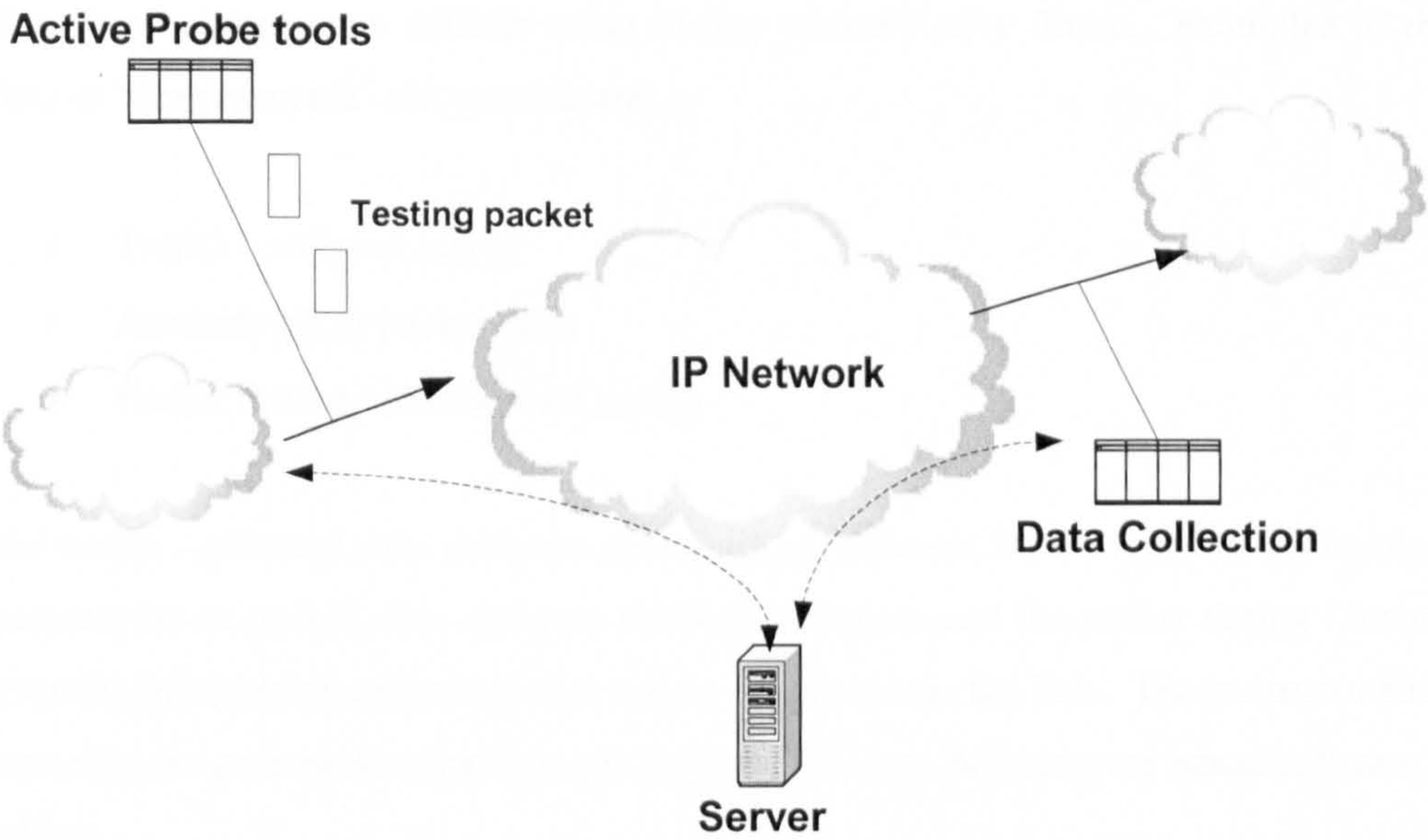


Figure 2-8 IPPM One Way Active Measurement

Figure 2-8 shows the OWAMP architecture. There are two messages defined in this protocol: OWAMP-Control and OWAMP-Test message. It is assumed that both end-points are being time synchronized. The client initiates a testing session by sending an OWAMP-Control message to the server. In the OWAMP-Control message, the length of the measuring period and the sampling frequency are specified. The Active Probe then generates the time-stamped testing packets to the remote reference point. At the remote point, the testing packets will be collected and the measurement results will be sent back to the server. Finally the client can fetch the result for analysis from the server if necessary. The difference between ICMP-based Active Measurement (Pings and Traceroute) and OWAMP is that the latter targets the one-way measurement (i.e. measuring the forward path), whereas, in the ICMP-based Active Measurement, both forward and return paths are involved.

2.3.2 Passive Measurement

Loosely speaking, the term Passive Measurement refers to those measurement methods that do not perturb the network load condition. In contrast to Active Measurement, no extra testing packets are generated during measurement and intrusion effect is introduced to the network. Passive Measurements are normally carried out by observing normal network traffic or monitoring the status

of network components without using testing packet/Active Probe. Examples of the information Passive Measurements[1] can provide are:

- Traffic / protocol mixes
- Accurate bit or packet rates
- Packet timing / inter-arrival timing

The traffic / protocol mix analyses the usage of different traffic types or protocols across a link. Accurate bit or packet rates analyses the link utilization and the packet timing / inter-arrival timing gives the information of inter-packet arrival timing across the link. The measurements are based on capturing the packets passing through the link. This can be hardware based/software based or MIB polling.

A. Hardware-based Passive Measurement

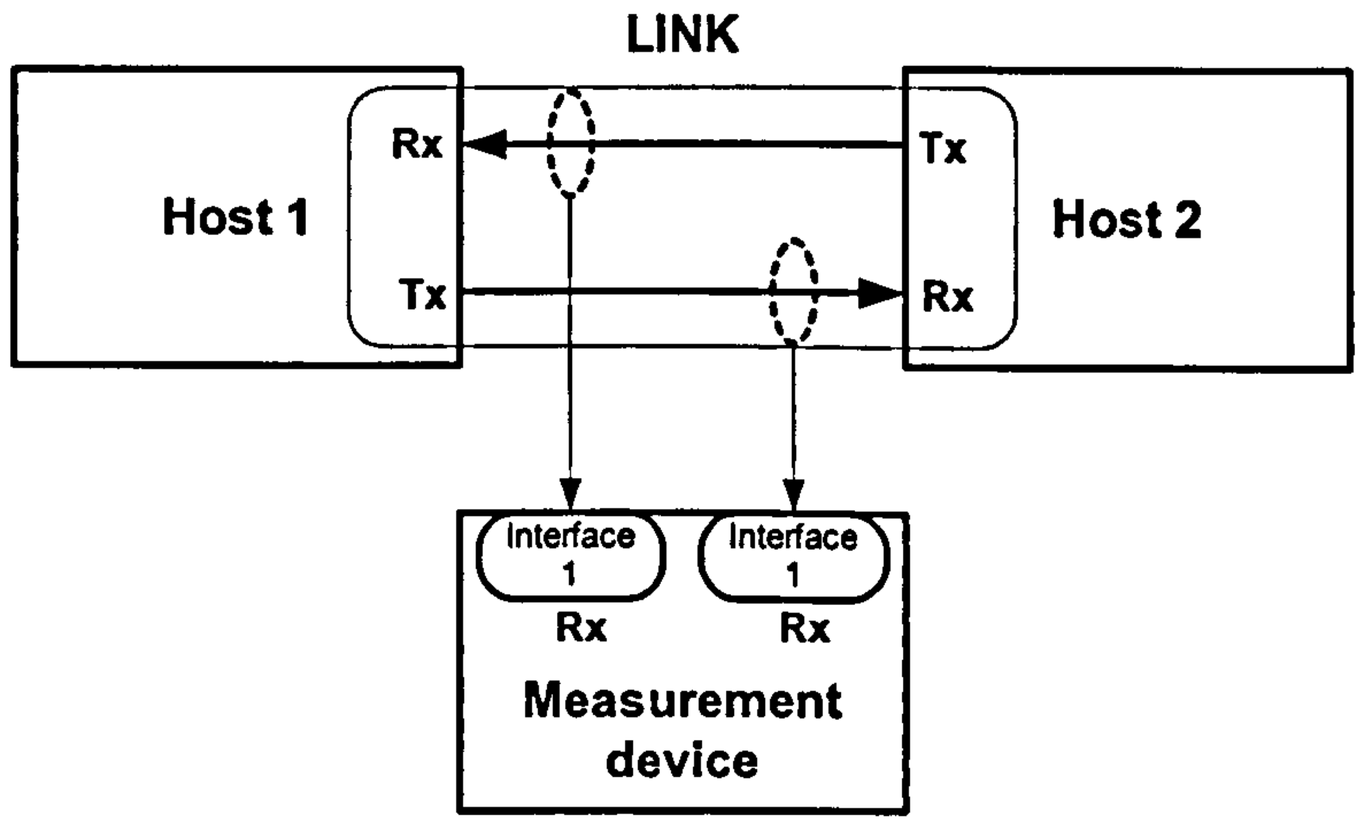


Figure 2-9 Passive Measurement (packet sniffing)

Figure 2-9 depicts the setup of an external Passive Measurement for packet monitoring. A part of the signals in the link passes to the measurement device. This can be achieved by using an optical splitter for fibre-optic network or resistor network for copper media. The packets passing through the link are captured by this measurement device to provide the information like; traffic / protocol mix, and the traffic volume in the link.

---

[1] In Chapter 4, we propose a Passive Measurement scheme by queue length monitoring for the end-to-end delay measurement.

B. Software-based Passive Measurement (Tcpdump)

Tcpdump is a software-based monitoring application, which is supported under most Unix systems. This program resides in the local node. It listens on a specified network interface, and captures all traffic seen on that link. Traffic monitors that rely on software-based applications like Tcpdump gain ease of use at the cost of performance. Design decisions, such as forcing all collection, analysis and display to be performed on a single machine, as well as implementation decisions, such as single-packet copies from the kernel level to user level, mean that this approach can not be scaled up to high network speed.

C. Polling MIB

Measurement tasks can be employed in the local node e.g. Queue Length Monitoring. The process, Queue length monitoring, has been adopted in schemes like Random Early Detection (RED) [FLO93] in which the queue length is being monitored for every arriving packet and this determines whether the incoming packet should be dropped or not, in order to avoid congestion. The statistics e.g. packet loss in this node/interface can be stored in the management objects like MIB. The MIB information is retrievable to the NOC with management tools like SNMP. This process is well known as MIB polling [CIS03]. Taking Junpiter's router's MIB definition as an example, the following MIB objects can be found [JUN02]:

- Queued packet/byte statistics
    - The total number of packets of specified forwarding class queued at the output on the given interface
- Transmitted packet/byte statistics
    - Number of packets of specified forwarding class transmitted on the given interface
- Tail-dropped packet statistics
    - The total number of packets of specified forwarding class dropped due to tail dropping at the output on the given interface
- RED-dropped packet statistics
    - Total number of packets belonging to the specified forwarding class dropped due to RED (Random Early Detection) at the output on the given interface

One of the MIB objects, Transmitted Packet, indicates the volume of the packets passing through the link, whereas, the object Tail-dropped or RED-dropped Packet statistics can be used to infer the packet loss in this local node when a FIFO queue or a RED queue are being used.

In this thesis, our measurement technique is also based on queue length monitoring for end-to-end delay performance measurement. The queue is passively monitored to infer the local delay distribution and so the end-to-end distribution. Since there are no testing packets in our measurement scheme, our scheme can be also considered as a passive (non-intrusive) measurement. This measurement scheme will be fully explained in Chapter 4.

2.3.3 Active and Passive Measurement Comparison

Table 2-2 outlines the features of Active Measurement and Passive Measurement. As we stated before, the main difference between Active Measurement and Passive Measurement is the presence or absence of the extra testing traffic. In Active Measurement, the testing packets pass through the network elements and the end-to-end path properties are measured. Without the testing packets, it is difficult for Passive Measurement to be employed in the end-to-end QoS related measurement. Nevertheless, owing to the favourable non-intrusive feature of Passive Measurement, it is attractive to explore the application of Passive Measurement in the end-to-end QoS related network measurement.

	Active Measurement	Passive Measurement
Configuration	Multi-point	Single or multi-point
Data size	Small	Large
Network Overhead	additional traffic	- Device Overhead - NO overhead if splitter or copper network is used
Purpose	Delay, packet loss, availability	Throughput, traffic pattern
CPU Requirement	Low to Moderate	High – traffic capture Low – polling MIB (like queue length monitoring)
Intrusion	***YES***	***NO***

Table 2-2 Comparison between network measurement approaches [HON03]

2.3.4 Network measurement-based Literature

In the previous sections, we discussed the network measurement methods and their application from the industry's point of view. Additionally, numerous measurement-based papers have been published and in this section, we will highlight those of interest.

A. Network Tomography

This topic addresses the difficulty, in the end-systems, of accessing the internal network element's characteristics. This problem can occur when the internal network element is owned by other network providers or in an inter-network domain. To solve the problem, the end-to-end measurement is performed to infer the internal network elements' characteristics, such as the loss characteristic [CAC99][PAD03] or the delay distribution in the intermediate node [PRE00][COA01]. This interesting research topic is referred to as network tomography. For the network tomography for the internal delay distribution inference, one of the approaches is to send a series of back-to-back testing packets to two different destinations. Using back-to-back packets means that two testing packets are sent one after the other from the Active Probe. The paths for these two back-to-back packets are assumed to be in a shape of a tree structure, i.e. the packets leaving from the Active Probe uses the same path as before to reach different destinations. Since these back-to-back packets are sent out in a short period of time, it is assumed that the delay experienced at the shared links will be the same, and so the difference between the two delay measurements can be attributed to the delays experienced on unshared links in the two paths [COA01]. This property is useful to infer the internal links' delay distribution by making the assumption of stationary delay distribution and independence of links' delay characteristic. In [COA01], a numerical iteration method called Expectation Maximization (EM) algorithm is used to solve this network tomography problem. This method induces the unobservable data (interior links' delays) based on the observable data (back-to-back end-to-end delay measurements).

B. Measurement-based Admission Control

The role of an admission control algorithm is to ensure that the admission of a new connection into a resource constrained network does not violate service commitments made by the network to the admitted flows, whilst maintaining good network utilization. One of the approaches to admission

control is known as parameter-based, which computes the amount of network resources required to support a set of flows given a priori flow characteristics. The new connection will not be accepted if the network resource is not sufficient. The main problem of parameter-based admission control is the difficulties of a priori flow characteristic estimation. It is difficult to accurately estimate the traffic parameters when a flow is first established. Inaccurate assumption of the traffic information may result in the service degradation to other existing flows. Guaranteed QoS performance can certainly be ensured by using worst-case traffic characteristic (e.g. using the peak rate), but this will surely lead to the link's under-utilization. Much research works have been carried out to study an approach called Measurement-Based Admission Control (MBAC). MBAC avoids this problem by shifting the task of traffic specification from the application to the network, so that admission decisions are based on traffic measurements instead of an explicit specification. MBAC consists of a measurement methodology, which prescribes what to measure and how, and a decision rule, which takes the measured data as input to predict how the current performance will be affected by the admission of a new flow. Measurements are usually on the level of traffic already existing in the network. [BRE00] outlines three measurement mechanisms as follows:

- Time-window:
    - This method is used in "Measured Sum" algorithm, see also [JAM97], the average incoming traffic is measured. The average load for every S sampling period is computed. Then, the highest average value from the last measurement windows T is used as an input for the admission control decision.
- Point Samples:
    - Compared with the Time-window, the measurement result of average load for every S sampling period is used instead.
- Exponential Averaging:
    - In this measurement mechanism, the measurement data is further smoothed by using exponential averaging. Similar to Time-window method, the average arrival rate is measured once every S sampling period. The final result used for admission control decision is computed by using infinite impulse response function with weight w to the measured data.

The measurement methods discussed above are passive ones. The traffic volume in a link is measured and interpreted in different ways. The measurement result is used for the admission control decision. Other different algorithms such as Measured Sum, Acceptance Region, Equivalent Bandwidth, Tangent at Peak, Tangent at Origin, Aggregate Traffic Envelopes are reviewed in [BRE00].

C. Bandwidth Adjustment

Bandwidth adjustments based on network measurement have been discussed in the works [MAL97], [KES99]. In [KES99], the author addresses the importance of the bandwidth adjustment. Owing to the large statistical complexity of traffic in broadband networks, connection admission controls can often only use crude initial estimates of the required bandwidth allotments. As a consequence, bandwidth adjustment is a methodology used to fine tune the allocated bandwidth in order to achieve the optimum network operation point. In [KES99], by exploiting the queue length distribution characteristic, it proposed two point queue length measurement and three point queue measurement for different traffic types such as short-range dependent (SRD) and long-range dependent (LRD) respectively. In this work, the queue length is passively monitored to measure the probability exceeding certain points. (Queue length monitoring is also employed in our work as discussed in Chapter 3 & 4) The measurement result is used to infer the queue length characteristic and then the effect of adding / deducing the allocated bandwidth on the performance is computed by using the sensitivity analysis[2]. This research work also inspired our work as discussed in Chapter 8. Compared with the work in [KES99], our approach is to estimate the traffic parameters from the measurement for the well-developed Excess-Rate analysis to determine the effect of the changing bandwidth. Unlike the work in [KES99], our work is also applicable to the scenario of changing bandwidth in great deal.

2.5 Summary

We have reviewed the network measurement techniques from different aspects: the motivation, different measurement approaches, features of the measurement methods and their applications. We have also highlighted the research which is network measurement-based. The motivation of network measurement comes from the network performance assessment or monitoring. Performing network measurement allows the operator to collect the useful data for analysis or assessment. A variety of network measurement techniques have been proposed and developed, but they can be classified into two broad classes: Active Measurement and Passive Measurement. Examples of Active Measurement tools are Ping, Traceroute or OWAMP, whereas, packet sniffing, tcpdump and polling MIB are typical examples of Passive Measurement tools. Intrusion effect and Heisenberg effect are two well-known unfavourable effects possessed in Active Measurement. Although the application of

[2] In [KES99], the amount of changing bandwidth is assumed to be so small that no busy period of the queue splits. Clearly, this assumption is invalid when the bandwidth alters in great deal.

Passive Measurement is limited, it does not cause any intrusion to the network. Therefore, it is advantageous to explore the usage of Passive Measurement in different applications.

Chapter 3 End-to-End Packet Delay Issues

In this chapter, we focus on the issue of end-to-end packet delay. To send a packet to the remote end, a packet will inevitably be delayed by different causes. We highlight various delay components and their characteristics. To assess or measure the packet latency performance, a variety of measurement techniques were developed and these are highlighted in this chapter. Their merits and weaknesses are compared. Queue length distribution estimation is our proposed method for the delay performance measurement, and is adopted in our measurement scheme. We review the related works which are based on queue length distribution estimation by measurement.

3.1 Why is End-to-End Delay Important?

In a heterogeneous data network, it is not unusual to carry different type of applications. Many applications may be delay-sensitive: i.e. the quality of the service is affected by the delay metrics. Table 3.1 shows examples of basic end-to-end delay requirement for some delay-sensitive applications.

	Application	Minimum user requirement
Inelastic applications	Video	5 frames per second
	Audio	Delay < 400ms
	Interactive real-time multimedia	Delay < 200ms Jitter < 200ms
Elastic applications	Web page access	Latency < 11 seconds

Table 3-1 Minimum latency thresholds for acceptable QoS [DEM00]

The end-to-end delay (latency) is one of the performance metrics characterizing QoS in a packet network [PAX98]. Taking the telephony service as an example, the speech quality is absolutely satisfactory when the end-to-end delay is less than 150 ms. The quality is fairly good when the latency is less than 250ms, but when the latency is more than 400ms, then the speech quality is totally unacceptable.

Various factors account for the end-to-end delay and these will be explained in next section.

3.2 Delay Components in the Network

The end-to-end packet delay is the time difference between the timestamps of departure at the source and arrival at the destination. Broadly speaking, the end-to-end delay can be divided into stochastic and deterministic components[3] as follows:

3.2.1 Processing Delay

Processing delay is the time needed to process a packet at each node and preparation for transmission. The processing delay is determined by the complexity of the protocol stack, the computational power available at each node and the link driver, which is determined by the available hardware rather than the traffic. Although, the processing delay is in general a stochastic random variable, its variation between the packets should not be significant especially in a regime of forwarding schemes like ATM or MPLS.

3.2.2 Transmission Delay

The time needed to transmit an entire packet, from first bit to last bit, over a communication link. The transmission delay is dictated primarily by the link capacity. If the packet size is fixed in the system such as ATM, then the transmission delay can be regarded as a fixed value.

3.2.3 Propagation Delay

The time needed to propagate a bit through the communication link. The propagation delay is deterministic, for instance, by the travel time of an electromagnetic wave through the physical channel of the communication path like fibre optic, or by electrical signal speed via the cable. It is independent of actual traffic on the link. The propagation delay can be significant as, for instance, in trans-Atlantic links or in Satellite link. In Satellite communications, the one-way propagation delays are around 110 – 150ms for medium earth orbit system (MEO), whereas, it can be as high as 250-

[3] In this thesis, the delay introduced in the network element only is considered. Those delay components introduced in the application level like packetization delay in real-time application are not considered as we are only focusing on network measurement, whereas, the application measurement is beyond our scope

280ms for geostationary satellites. Under these circumstances, there may be higher fixed delays than the stochastic delays.

3.2.4 Queuing Delay

This is the waiting time of packets needed in the buffer of routers before transmission. Queuing delay occurs when the queue service rate is not fast enough to serve all the incoming packets from different sources using this queue. Therefore, the packets accumulate in the queue and wait for the service. The queuing behaviour depends on the traffic pattern and statistical nature of the arrival and service process. The queuing delay in a node may vary significantly from packet to packet due to the randomness of the incoming packets' arrival. This is the major cause of the packet delay variation and is typically stochastic in nature. For example, in a FIFO queuing system, the packets will experience longer queuing time when seeing a longer queue at the arrival instance.

The above delay components occur when a packet traverses a node. The *per-hop delay* experienced by a packet is the sum of the above delay components. The *end-to-end delay* will then be the sum of the overall *per-hop* delay introduced to the packet by each hop along the end-to-end path traversed by the packet.

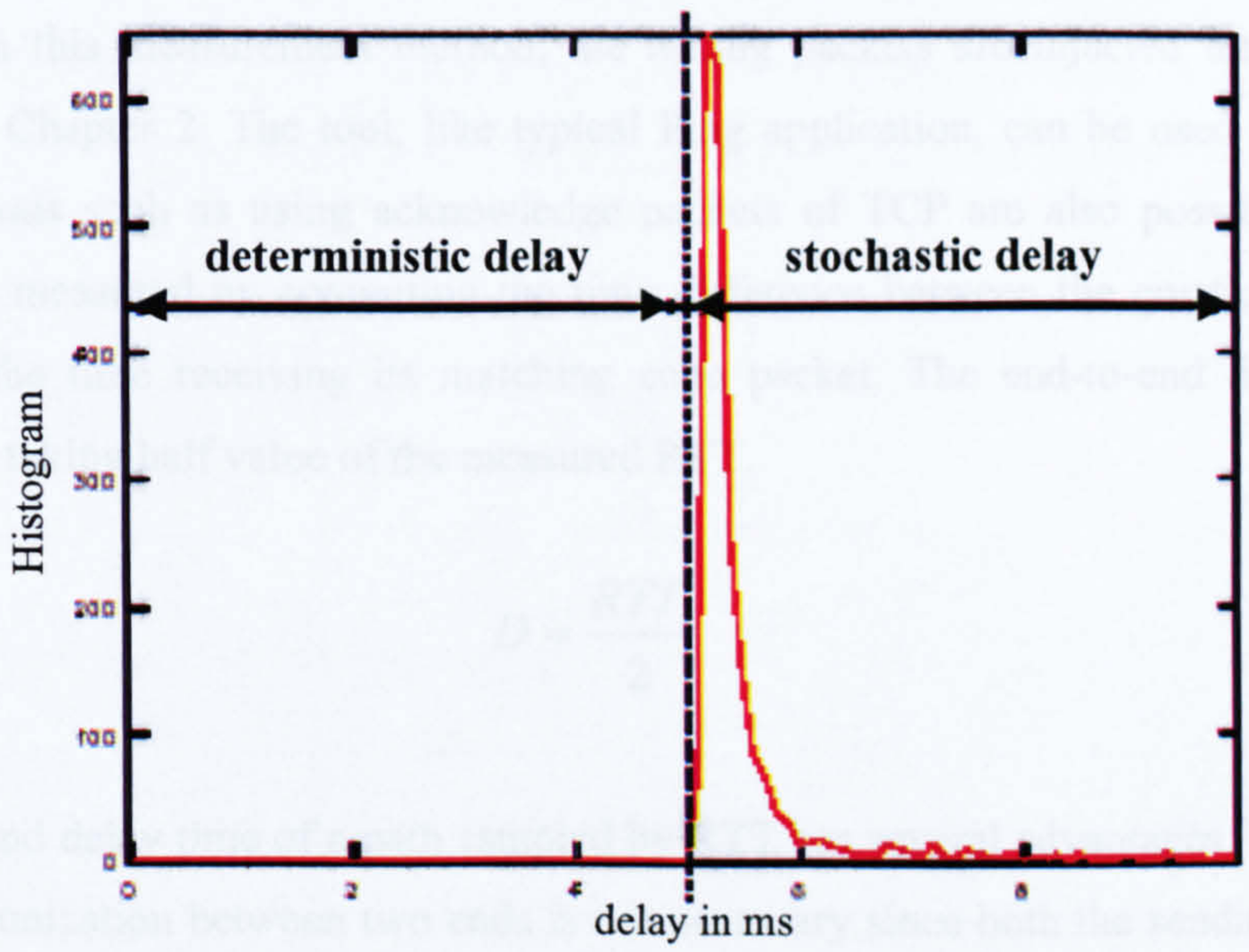


Figure 3-1 A typical delay histogram (adapted from [BOV02])

Figure 3-1 depicts a typical delay histogram plot. It is noted that the plot consists of two regions such as deterministic delay and the stochastic delay. Since the propagation delay is invariant and fixed along a specific path, this contributes to the deterministic delay. As discussed in Section 3.2.1 and 3.2.2, the processing delay and the transmission delay can be stochastic in nature however, the queuing delay is relatively more significant. As a result, we consider the queuing delay as the main source for the stochastic component. Similar assumptions can be found in the works [YAT93][RIB00] that the propagation delay and the packet processing time are ignored. With this assumption, the queue length distribution is highly correlated with the delay distribution. This will be further elaborated upon in Section 3.3.C.

3.3 End-to-End Delay Measurement

The general review of network measurement is given in Chapter 2. In this section, we further discuss the different network measurement methodologies for the end-to-end delay measurement. Their features are discussed and compared.

A. Estimation of End-to-End Delay by RTT Time

The end-to-end delay measurement by using Round Trip Time (RTT) is prevalent because of its simplicity. In this measurement method, the testing packets are injected into the testing path as discussed in Chapter 2. The tool, like typical Ping application, can be used for this purpose, and variant methods such as using acknowledge packets of TCP are also possible. The RTT is then sampled and measured by computing the time difference between the creation time of the testing packet and the time receiving its matching echo packet. The end-to-end delay D for a path is estimated by taking half value of the measured RTT.

$$D = \frac{RTT}{2} \qquad \textbf{Equation 3-1}$$

The end-to-end delay time of a path sampled by RTT has several advantages. With this method, the clock synchronization between two ends is not necessary since both the sending time and receiving time are referenced at the sending end. Secondly, measurement tools such as ICMP-based Ping can be used and they are widely available in many platforms and equipments. Finally, there is no need to implement any measurement equipment in the remote end. This facilitates the operators to perform the measurement.

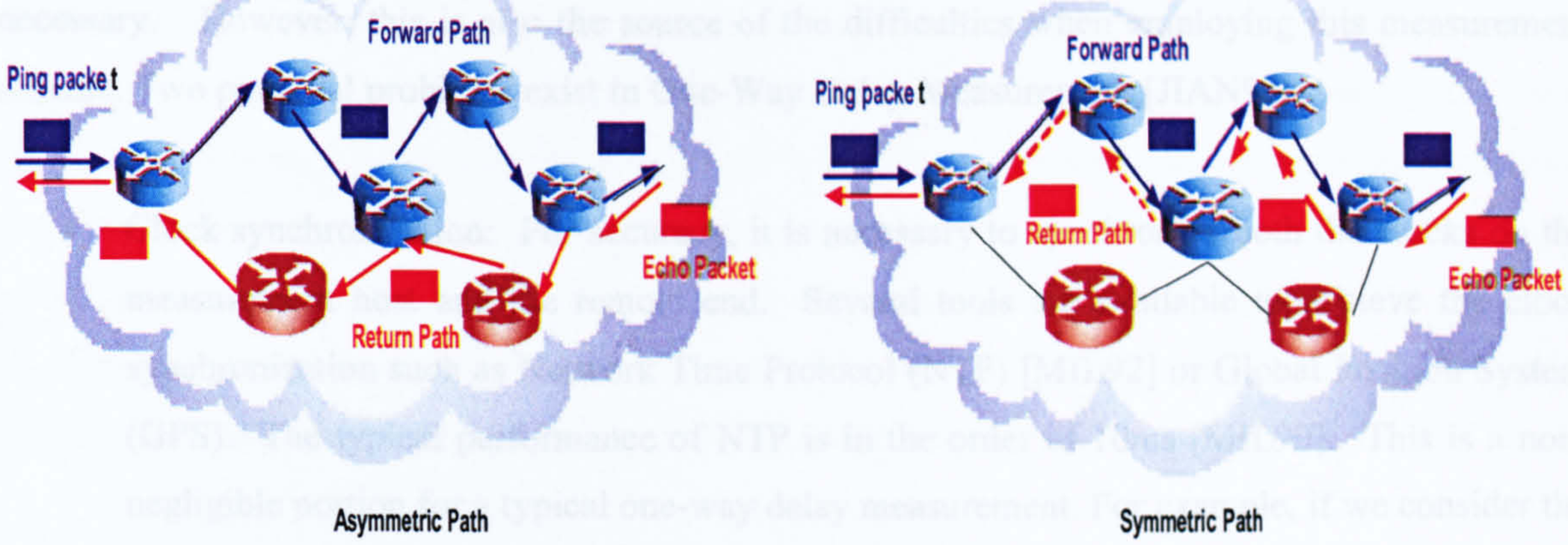


Figure 3-2 Asymmetric path and symmetric path

This measurement method is a branch of Active Measurement due to the existence of testing packets. This measurement scheme suffers from Active Measurement's unfavourable effect: network intrusion and the Heisenberg effect. Besides, there are several shortcomings in this scheme. Clearly, the estimation by taking half of the RTT time can be very coarse and inaccurate. The echo packet may take a different path with what the Ping packet traverses. The asymmetric path phenomenon is shown in Figure 3-2. [CLA93] [PAX96] point out that network topologies are often asymmetric. Owing to the asymmetric path, the times taken for traversing the forward and return path are different and this may lead to poor estimation in Equation 3-1.

Even if the echo packet indeed takes the same path as the Ping packet, the symmetric path does not guarantee the symmetric delay between the forward and return path. Some of the links in the return path may be in a congestion status that may cause more delay time or vice versa.

As a result, it can be noted that the latency estimation by measuring RTT is very inaccurate indeed although it is easy to perform the measurement.

B. One-Way Delay Measurement

One-way measurement is outlined in Section 2.3.1.1.c. As discussed, this measurement scheme samples the one-way (end-to-end) properties by sending the testing packets. The measurement data are collected at the remote end. Since there is no return path involved, the one-way property is measured directly at the receiving end, and the measurement result is more accurate compared with RTT measurement.

In One-way Delay Measurement, clock synchronization between the host and the remote end is necessary. However, this is also the source of the difficulties when employing this measurement scheme. Two potential problems exist in One-Way Delay Measurement [JIAN99]:

- Clock synchronization: For accuracy, it is necessary to synchronize both the clocks on the measurement host and the remote end. Several tools are available to achieve the clock synchronization such as Network Time Protocol (NTP) [MIL92] or Global Position System (GPS). The typical performance of NTP is in the order of 10ms [MIL90]. This is a non-negligible portion for a typical one-way delay measurement. For example, if we consider the propagation delay to be a main contributor of one-way delay, then a US coast-to-coast delay is around 30ms. GPS is a U.S. satellite-based navigational system that provides both location and time services. Compared with NTP, GPS is more precise. In GPS, these precise external time sources result in a clock synchronized to real time with an offset of few hundred nanoseconds [ADV99]. Its main drawback is the cost of equipment and its deployment. In addition, there can be logistical difficulties in placing GPS antenna [LUC01].
- Clock drift: Even if clocks are initially synchronized, there will be a slow shift of time between two clocks of the host and the remote end because they operate at slightly different frequencies. Since no two clocks are identical, the clock drift is inevitable. If the clock is driven by a crystal oscillator, the frequency of the crystal oscillator depends on its shape, size, temperature, etc. Therefore, the drift rate remains mostly constant when surrounding conditions (such as temperature) are stable. Typical drift rates of crystal oscillators are on the order of 100 μs per second [DAL99].

C. Distributed Measurement

In the previous sections A and B, the measurement schemes are based on Active Measurement. The prime weaknesses of Active Measurement are discussed in Chapter 2. Apart from the intrusive effect and the Heisenberg effect (see Section 2.3.1), another downside of using active probing for end-to-end measurement is that the testing packet may receive different treatment to the actual traffic [BRO01], which also leads to a wrong measurement result.

In contrast to Active Measurement, Passive Measurement has neither an intrusive effect nor an "Heisenberg" effect. Although the application of Passive Measurement is still limited, it is interesting to investigate its application to end-to-end delay performance measurement.

In Section 3.2, we explained various delay components existing in the network domain. This consists of the deterministic and the stochastic delay. The propagation delay contributes to the deterministic delay. For the stochastic delay, the queuing delay is the main cause of this delay behaviour. Although the processing delay and transmission delay may cause a small delay variation from packet to packet, in this thesis, we assume both factors are less significant compared with the queuing delay [YAT93][RIB00]. Since the variant of the packet delay is assumed to be due to queuing delay, the end-to-end delay is the sum of the total of queuing delays experienced at each hop and a constant delay time caused by the deterministic delay

Several factors affect the queuing delay. Certainly, the queuing discipline is a key factor affecting the queuing behaviour. In this thesis, we choose to focus on switches/routers with First-In-First-Out (FIFO) queuing. This is because FIFO is extremely efficient and easily implemented. It is also already used in a variety of switching/routers equipment [HAO98]. Weighted Round Robin (WRR) and Weighted Fair Queuing (WFQ) are also considered when different classes of traffics are supported.

With a FIFO queuing discipline, the incoming packet needs to wait for the service until all the existing packets in the queue are served. Therefore, the queuing time is closely related to the queue length seen by the incoming packets. In other words, the queue length seen by an arrival reflects the queuing time that the incoming packet will experience. As a consequence, the queue length distribution is expected to closely match with the per-hop delay distribution. *The per-hop delay distribution is just a scaled shifted version of the queue length distribution.*

In this thesis, our approach is to capture every per-hop queue length distribution by Passive Measurement, more precisely, by queue length monitoring. The measurement method will be discussed in Chapter 4 in detail. The queue length distributions of every queue along a path are used to deduce the end-to-end delay distribution. This can be achieved by convolving the queue length distributions of the node along the path[4]. In this process, we will have both the knowledge of per-hop queuing distributions and end-to-end queuing distribution. Finally, the end-to-end delay distribution will then be determined by adding the deterministic components to the scaled end-to-end queuing distribution.

In [JIA99], authors point out that the local (per-hop) distribution is essential to detect possible QoS degradation and to locate the cause of the degradation. The end-to-end measurement is not sufficient to locate the cause of the degradation. If only the end-to-end distributions are known, it is not

[4] The justification of using convolution for end-to-end delay distribution estimation is discussed in Section 4.3

possible to determine the degradation in each segment unless some sophisticated mechanisms are employed e.g. network tomography as discussed in Chapter 2 [COA01]. Therefore, apart from estimating the end-to-end delay distribution, the per-hop (local) queue length measurement can also be used for the detection of QoS degradation. For example, in Figure 3-3, the access router becomes the bottleneck of the path. This results in a relatively longer tail in queue length distribution. So, the bottleneck is identified by knowing each queue length distribution.

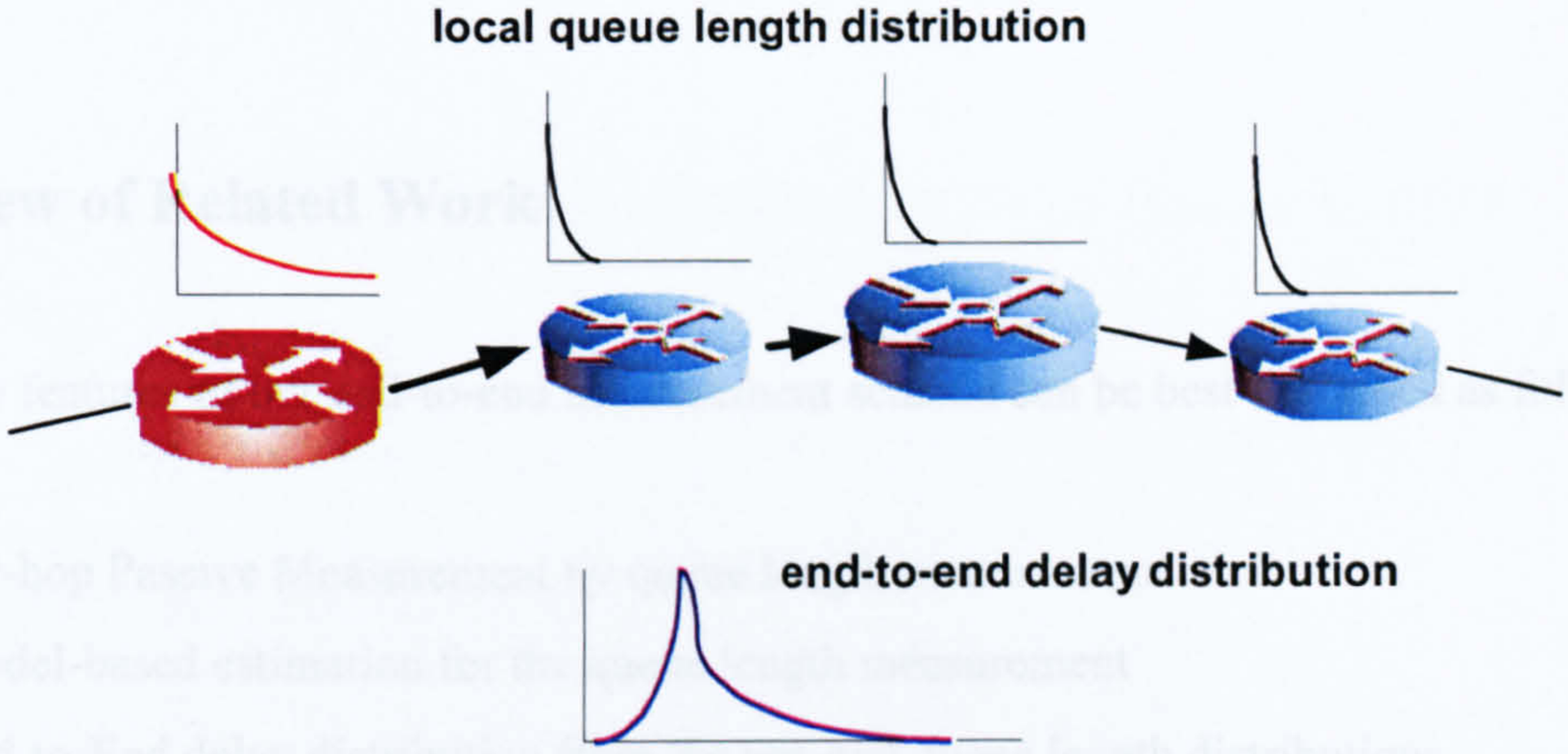


Figure 3-3 Per-hop queue length distribution and end-to-end delay distribution

Another advantage of using distributed measurement is its scalability. In [CHA00], authors point out that it is impractical to measure or collect performance measurements of each flow from every network device. This approach may be reasonable for a small network, but it is inefficient and not scalable.

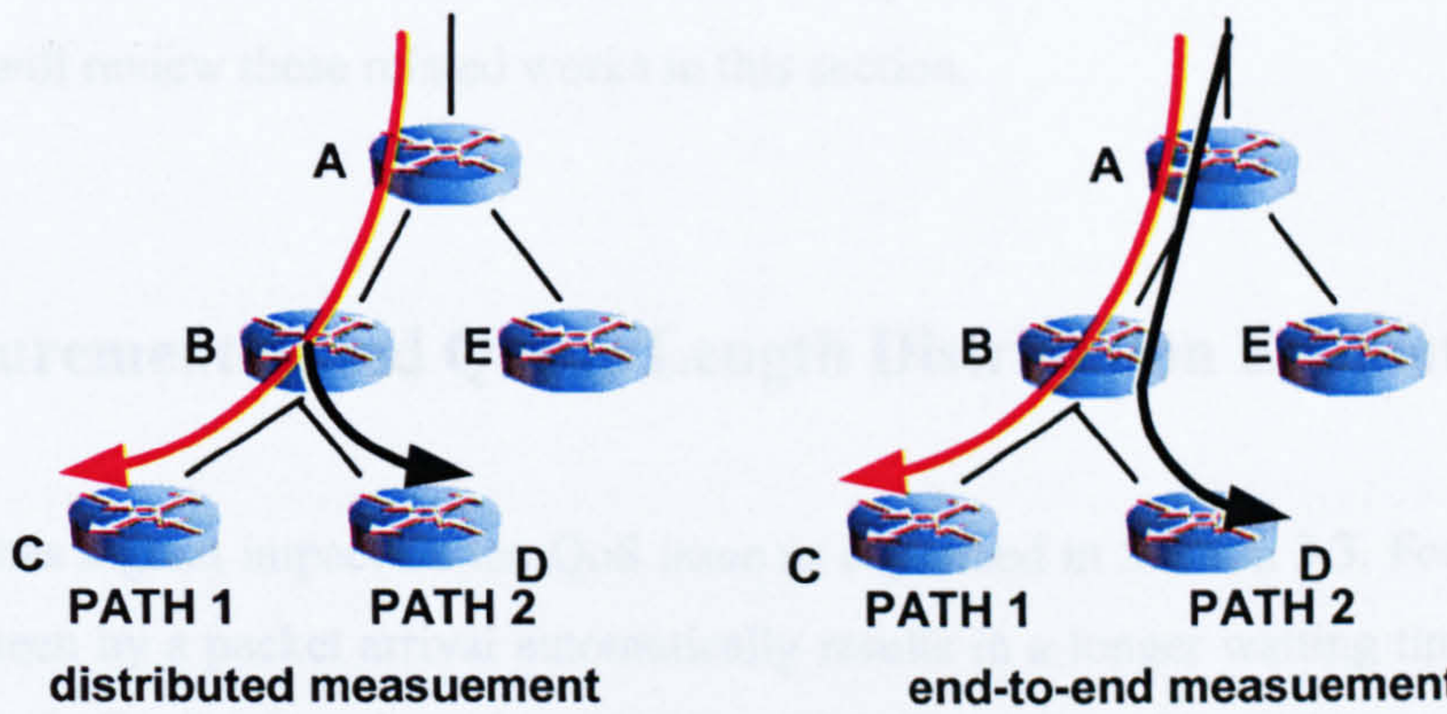


Figure 3-4 Distributed measurement and end-to-end measurement

In Figure 3-4, a part of the network is represented with a tree topology. Generally, some routes share the same path before diverging to their destination. In this case, route 1 and route 2 share a

common path which includes router A and router B. The measurement results (per-hop queue length measurement) obtained in router A and B can be used to deduce the end-to-end delay distribution for both route 1 and route 2. In other words, the measurement result can be *re-usable* as long as it is involved in the path of interest. On the other hand, for the end-to-end measurement, two separate measurements are necessary even though there exists a common path between two routes[5]. Measurement can be costly. Therefore, it is important to keep the measurement cost to be minimal to ensure the revenue.

3.4 Review of Related Work

Several key features of our end-to-end measurement scheme can be best described as follows:

- Per-hop Passive Measurement by queue length measurement
- Model-based estimation for the queue length measurement
- End-to-End delay distribution from the per-hop queue length distributions

In our end-to-end measurement methodology, per-hop queue length monitoring is involved. The measurement result is used to infer the per-hop queue length distribution with model-based estimation. Finally, the end-to-end delay distribution can be determined based on per-hop queue length distributions along the path. Our methodology will be presented in Chapter 4. There are a variety of related research works in the area of "queue length monitoring", "model-based queue length distribution estimation" and "end-to-end delay measurement". Prior to discussing our approach, we will review these related works in this section.

3.4.1 Measurement-based Queue Length Distribution Estimation

Queue length has a great impact on the QoS issue as explained in Section 3.3. For a FIFO queue, a longer queue seen by a packet arrival automatically results in a longer waiting time for this packet. The queue length distribution gives the statistics and information of how the per-hop packet delay

[5] Taking a mesh network with 4 fully connected nodes as an example, the number of possible paths between a pair of nodes is 5. Hence, the number of measurement for direct end-to-end measurements for all possible pairs is 60. On the other hand, as there are only 6 links, by considering both forwath and return path in a link, 12 measurements are needed when performing distributed measurement. When the number of nodes in the mesh network increases to 5, then, the number of measurement for direct end-to-end measurement increases to 320, whereas, the number of measurement increases from 12 to 20 by using distributed approach.

distributes. Additionally, the queue length distribution is an effective means to infer the packet loss in a hop especially if the packet loss event is rare (e.g. the target packet loss in ATM may be as small as 10^{-9} [LIU02]).

There are two major approaches for measurement-based queue length estimation: direct queue length monitoring and the traffic pattern measurement.

3.4.1.1 Queue Length Distribution Estimation by Queue Length Monitoring

A. Histogram Representation

Research into direct queue length monitoring can be found in [MAL97][SIL98][KES99]. In [SIL98], a histogram is used to represent the queue length distribution when a FIFO queue is monitored. The frequency of occurrence of different bins b_l as shown in Figure 3-5 is recorded. By using a histogram to represent a queue length distribution, this alleviates the measurement storage requirement, reduce the amount of data transfer between the measurement node and the NOC. The bin b_k is considered to be hit, if the queue length Q_t seen by arrivals is less than or equal to b_k. The estimate for the queue length distribution from the n empirical measurement is defined as follows:

$$P\{Q_t \le b_k\} \approx \frac{1}{n}\sum_{i=1}^{n} 1\{Q_{ti} \le b_k\} \qquad \textbf{Equation 3-2}$$

where $1\{\cdot\}$ denotes the identity function.

The authors argue that the queue length distribution can be well-modelled by the sum of a series of exponential functions for Markov-modulated traffic. The prime objective of the work [SIL98] is to estimate the parameters of the model from the measurement result with some Model Selection Algorithms. The estimate of the delay distribution is used to infer the QoS performance.

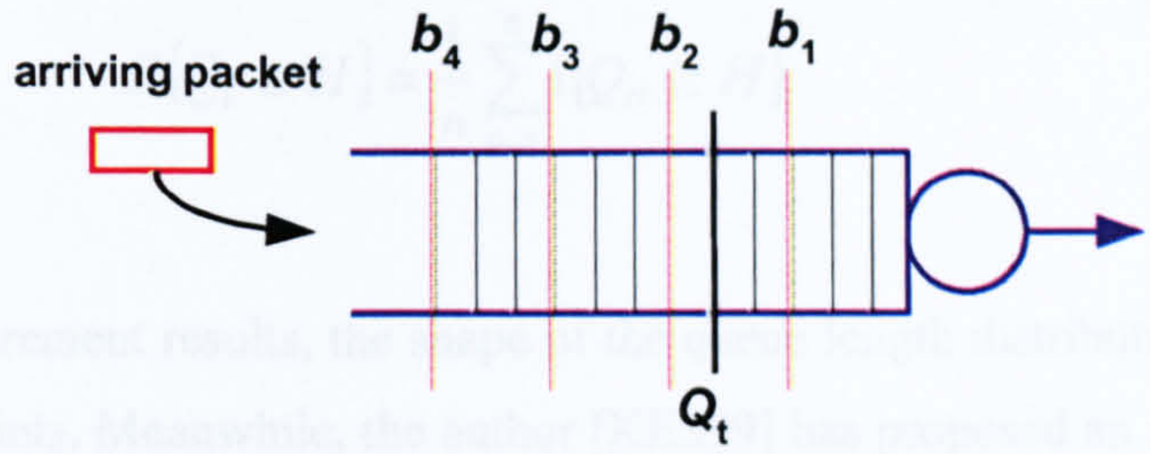


Figure 3-5 Queue length monitoring method in [SIL98]

B. Interpolation Method

Queue length monitoring can also be found in the works [MAL97] [KES99]. Unlike the queue monitoring scheme in [SIL98], the queue region is separated into two or three regions respectively in the work of [MAL97] [KES99] as shown in Figure 3-6. In these papers, the authors argue that the common queue length distribution can possess a geometric tail. It is well known that the tail of the distribution exhibits exponential decaying for a wide class of traffic models. Under this circumstance, the shape of the queue length distribution can be captured by performing two point queue length measurement.

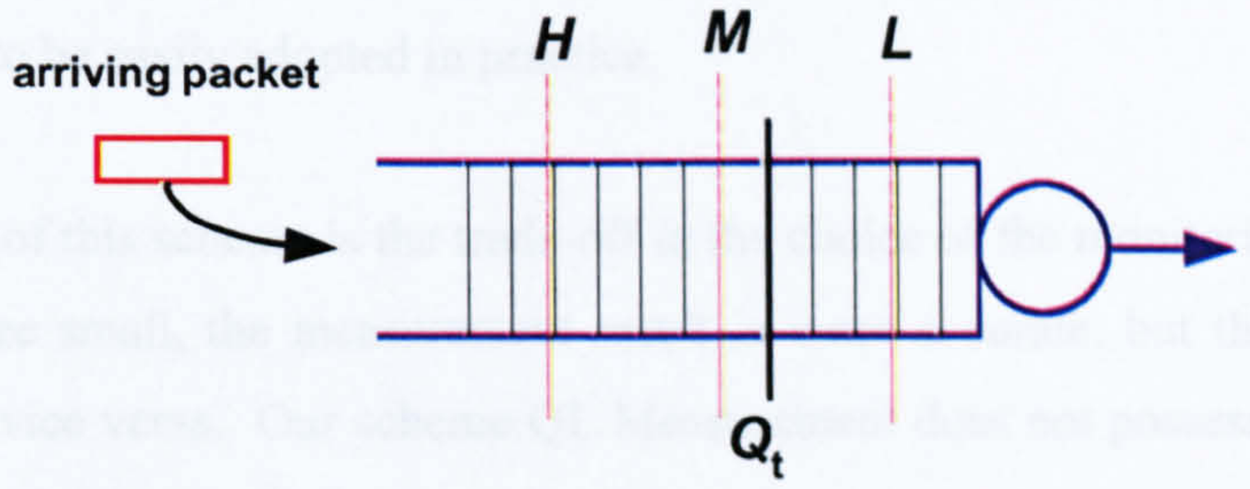


Figure 3-6 Queue length monitoring method in [KES99]

In the two point measurement, there are two pre-assigned points such as L and H. The frequency of occurrence of the queue length seen by arrivals greater or equal to L and H are measured from n observations.

$$P\{Q_t \geq L\} \approx \frac{1}{n}\sum_{i=1}^{n} 1\{Q_{ti} \geq L\} \qquad \textbf{Equation 3-3}$$

$$P\{Q_t \geq H\} \approx \frac{1}{n}\sum_{i=1}^{n} 1\{Q_{ti} \geq H\}$$ **Equation 3-4**

With the above measurement results, the shape of the queue length distribution can be estimated by extrapolating these points. Meanwhile, the author [KES99] has proposed an alternative approach for the case of long-range dependent traffic. Long-range dependent property has been observed in some highly-correlated traffic such as video and data. In the presence of long-range dependent traffic, the queue tail distribution will be heavy-tailed. One of the possible models of this is known as the Weibullian distribution. In this case, two-point measurement is not sufficient to characterize the shape of the Weibullian distribution. Therefore, an extra point *M* is introduced for monitoring. Similar to the measurement for point *H* and *L*, the frequency of occurrence of the queue length seen by arrivals greater than *M* will also be recorded. The shape of the queue length distribution can then be inferred by these three measurement results. The objective of estimating queue length distribution is to provide information for the sensitivity (or perturbation) analysis in case of the changing bandwidth. The author [KES99] has addressed the importance of this perturbation analysis, since it is hard to define the optimum bandwidth allocation and so the perturbation analysis is performed to fine-tune the allocated bandwidth.

The author [KES99] has stressed that the direct empirical measurement requires the least amount of computation and is feasible at very high transmission speeds. This allows the queue length monitoring scheme to be easily adopted in practice.

The main drawback of this scheme is the trade-off in the choice of the monitoring points. When the monitoring points are small, the measurement result is more accurate, but the extrapolation error becomes larger and vice versa. Our scheme QL Measurement does not possess this drawback. This will be explained in Chapter 4.

3.4.1.2 Queue Length Distribution Estimation by Traffic Measurement

In the previous sections, we highlighted the techniques for queue length distribution estimation by direct queue length monitoring. An alternative approach based on traffic measurement has also been reported. With this approach, the statistic of the traffic passing through a link is measured instead of monitoring the queue length. The measurement result is then used to infer the queue length

distribution with analytical models such as Large Deviation Theory (LDT) [DUF98] [CHAN00] and Maximum Variance Theory (MVT) [CHO97].

A. Large Deviation Theory

The Large Deviation Theory has been studied in the application of *rare event* analysis. For instance, ATM is a packet switching standard that has been proposed as having cell loss rates in the order of 10^{-9} to 10^{-12} [LIU02].

A queuing analysis based on the Large Deviation Theory reveals that, for wide-ranging traffic input, the tail of the queue length distribution decays exponentially in the regime of large buffers. The rate of decay is known as the decay rate and the offset is the decay constant. The decay rate can be estimated by using the arrival rate function using the knowledge of the service rate. Traffic measurement is used to estimate the cumulant generating function of the arrival process. The measurement method is outlined as follows:

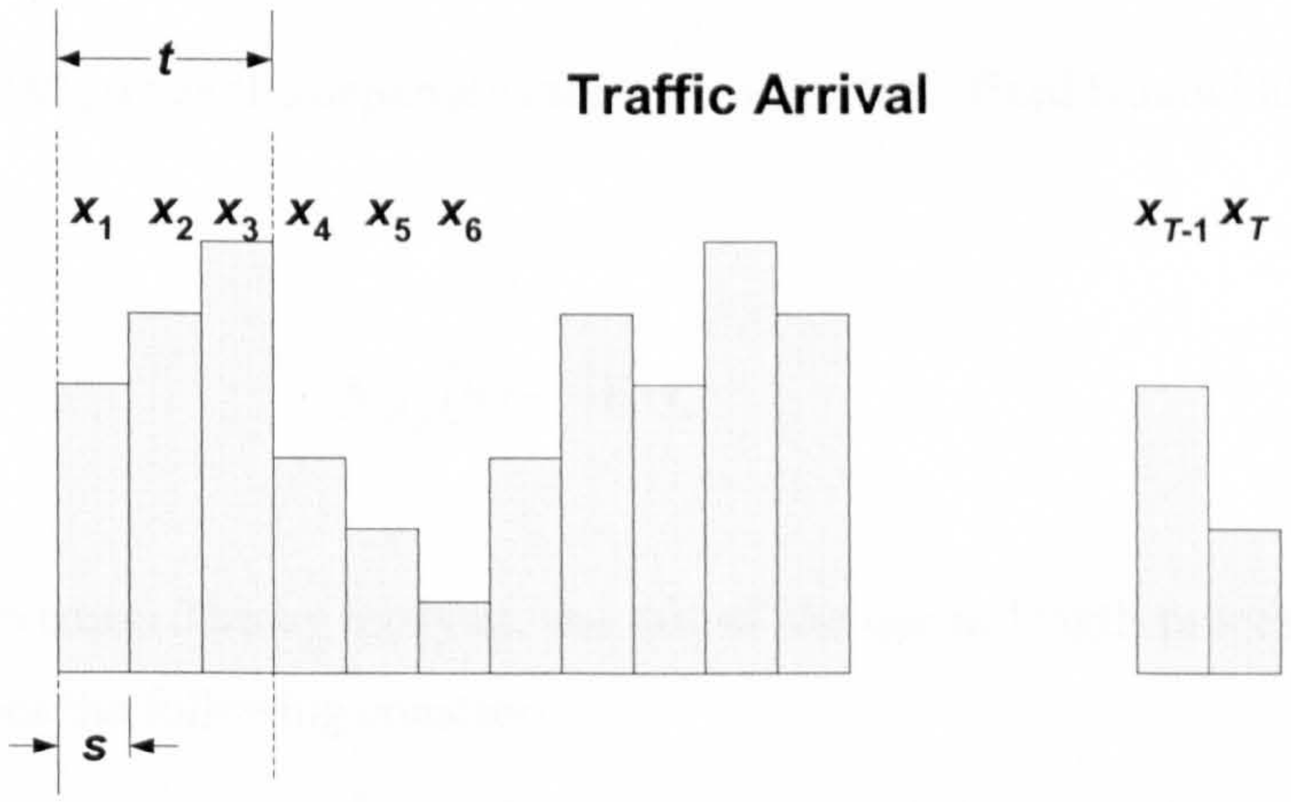


Figure 3-7 The traffic volume measurement

Figure 3-7 depicts the traffic volume measurement in [DUF98]. In each time slot *s*, the traffic volume (in terms of either number of packets or total number of received bits) is measured and recorded. This can be achieved by employing passive measurement as discussed in Chapter 2.

The traffic sample $(X_i)_{i=1,2,\ldots T}$ is further grouped and the measurement window is divided into n blocks of length t such that the aggregate arrival $Y_{j,t}$ over time t is defined as follows:

$$Y_{j,t} = \sum_{i=(j-1)t+1}^{tj} X_i \qquad \textbf{Equation 3-5}$$

The cumulant generating function for the traffic arrival $\Lambda_{A,t}(\theta)$ is estimated as follows:

$$\Lambda_{A,t}(\theta) = \frac{1}{t}\log\left(\sum_{i=1}^{n} e^{\theta Y_{i,t}} \Big/ n\right) \qquad \textbf{Equation 3-6}$$

Likewise, the cumulant generating function for traffic departure $\Lambda_{B,t}(-\theta)$ is given as:

$$\Lambda_{B,t}(\theta) = \frac{1}{t}\log\left(\sum_{i=1}^{n} e^{-\theta Z_{i,t}} \Big/ n\right) \qquad \textbf{Equation 3-7}$$

where $Z_{i,t}$ denotes the aggregate departure amount for period t of sample i.

In a special case G/D/1, when the departure rate is constant for a fixed bandwidth C, then Equation 3-7 becomes:

$$\Lambda_{B,t}(\theta) = -\theta \cdot C \qquad \textbf{Equation 3-8}$$

From the Large Deviation Theory analysis, the tail of the queue length process for wide-range of traffic model satisfies the following equation:

$$\lim_{x\to\infty} \frac{1}{x}\log P[Q \geq x] = -\theta^* \qquad \textbf{Equation 3-9}$$

where $\theta^* > 0$ is the largest root of the equation

$$\Lambda_A(\theta) + \Lambda_B(-\theta) = 0 \qquad \textbf{Equation 3-10}$$

where $\Lambda_A(\theta)$ and $\Lambda_B(-\theta)$ are the cumulant generating function as defined as in Equation 3-6 and Equation 3-7 respectively that can be estimated from measurement.

Then, the queue length process is estimated as:

$$P[Q \geq x] \approx e^{-\theta^* x} \qquad \textbf{Equation 3-11}$$

Equation 3-11 can be used to estimate the overflow probability in a queue with a large finite buffer of size x. If we are interested in the queue length distribution, i.e. $P[Q=x]$, then it can be estimated from Equation 3-11 as follows:

$$P[Q = x] = \frac{d}{dx}(1 - P[Q \geq x]) \approx \theta^* e^{-\theta^* x} \qquad \textbf{Equation 3-12}$$

With the expression of Equation 3-12, it can be noted that the queue length distribution decays exponentially. In a log-linear plot, the queue length distribution forms approximately a straight line with offset c and the slope $\log\eta$. The symbol c and η are the decay constant and decay rate which are equal to θ^* and $e^{-\theta^*}$ respectively.

There are several issues with this measurement method. This measurement method may cause storage problems if larger sample points are collected for smaller t in Equation 3-5. However, for the larger t, fewer sample points will be obtained and this leads to increased sample variance. The queue length distribution estimated by Large Deviation Theory is an asymptotic one, therefore, the tight estimation of the queue length distribution is not guaranteed.

B. Maximum Variance Theory

Large Deviation Theory reveals that the queue length distribution exhibits exponential decay for a wide-range of traffic models. However, the queue length distribution may have power-law decay in the presence of long-range-dependent traffic. Recent experimental results revealed that the traffic in many data networks exhibits this long-range dependent property [KLI94] [LEL94] [CRO95]. Compared with the short-range-dependent traffic, long-range-dependent traffic is more bursty and has a substantial correlation function even in the long time lag.

Owing to the different shape of the queue length distribution, the Large Deviation Theory may not be applicable for the queue length process analysis. A novel technique called Maximum Variance Theory (MVT) was developed by [CHO97]. This technique is motivated by the fact in that the

advent of high-speed networks, a single link will carry hundreds or even thousands of applications. This results in a very natural application of the Central Limit Theorem, to model the network traffic by a Gaussian process. Then, the arrival process can be completely specified by its first two moments, the mean and the auto-covariance function. The queuing behaviour is then analysed through a fluid queuing model.

Due to using a Gaussian process, the tail of the queue length distribution is given in the form of:

$$P(\{Q > x\}) \approx \psi\left(\frac{x - \kappa_{t_x} \cdot t_x}{\sqrt{\nu_{t_x}}}\right) \qquad \textbf{Equation 3-13}$$

where $\psi(\cdot)$ denotes the standard Gaussian tail function i.e. $\psi(\varpi) = \frac{1}{\sqrt{2\pi}} \int_{\omega}^{\infty} e^{\frac{-z^2}{2}} dz$, and κ_{t_x} and ν_{t_x} is the mean and variance of the increment with respect to time t_x in the fluid-flow model.

This technique was inspired by the concept "rare event happens in the most likely way". Based on this concept, the time-scale t_x with respect to queue length x that maximize the factor $\nu_{t_x} / \left(x - \kappa_{t_x} \cdot t_x\right)^2$ also maximizes the probability in Equation 3-13. This specific time-scale is called the dominant time-scale in [EUN03].

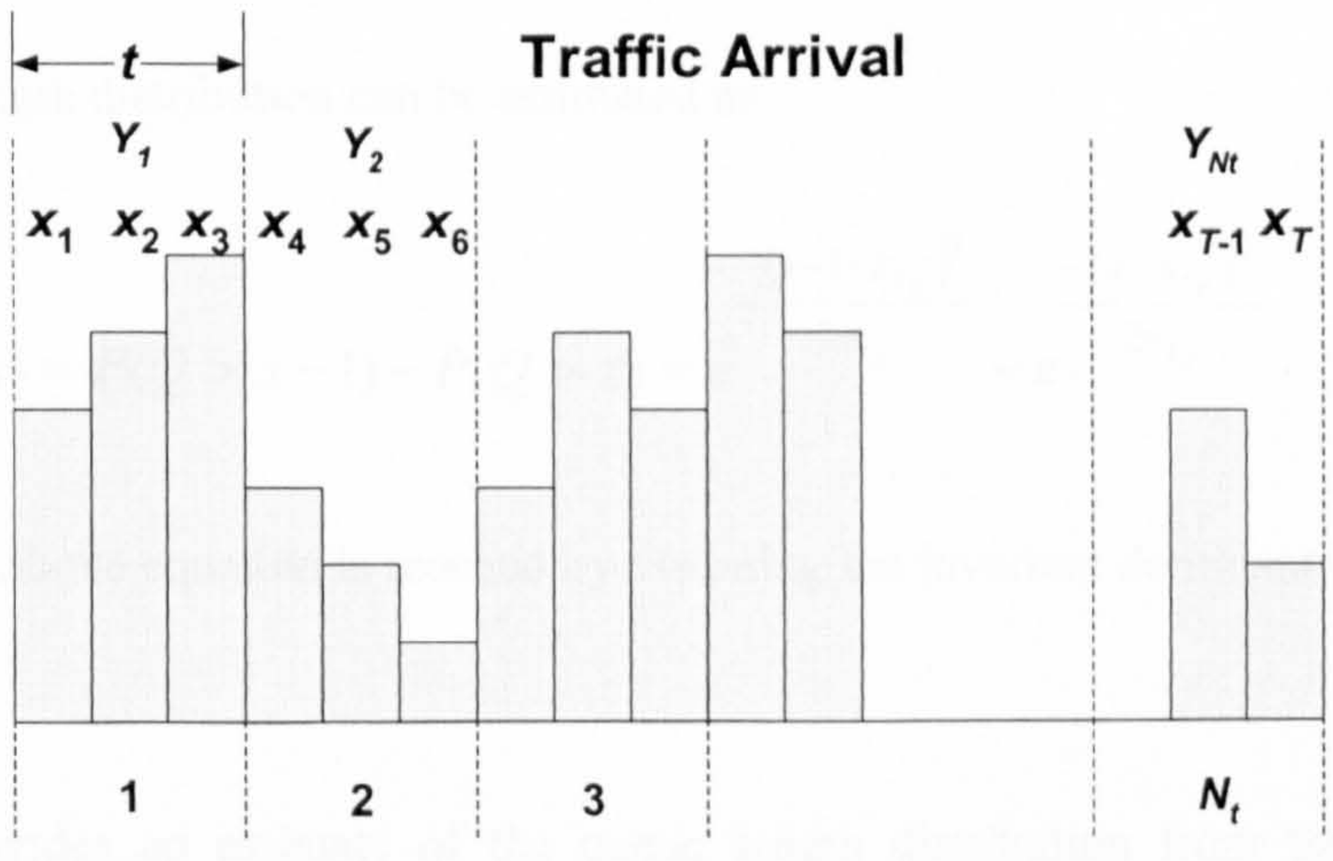


Figure 3-8 The traffic volume measurement for MVT

Figure 3-8 depicts the traffic volume measurement for MVT. The traffic is recorded over a small time granularity. Similar to Figure 3-7, the successive time granularity is grouped into a larger time scale t and N_t aggregate traffic Y_i samples are obtained. Then, the mean κ_t and variance ν_t with respect to time scale t are estimated from the empirical measurement as follows:

$$\kappa_t = \frac{1}{t \cdot N_t} \sum_{i=1}^{N_t} Y_i - C \qquad \textbf{Equation 3-14}$$

$$\nu_t = \frac{1}{N_t - 1} \left\{ \sum_{i=1}^{N_t} Y_i^2 - \left(\frac{1}{N_t} \sum_{j=1}^{N_t} Y_j \right)^2 \right\} \qquad \textbf{Equation 3-15}$$

Then the dominant time scale t_x with respect to a specific queue length x is the time scale that maximizes the following expression:

$$t_x = \max_t \frac{\nu_t}{(x - \kappa_t \cdot t)^2} \qquad \textbf{Equation 3-16}$$

[CHO97] points out that the tail probability can be accurately approximated by the following equation:

$$P(Q \geq x) \approx e^{\frac{-(x-\kappa_{t_x})^2}{2\nu_{t_x}}} \qquad \textbf{Equation 3-17}$$

Then the queue length distribution can be estimated as

$$P(Q = x) = P(Q > x-1) - P(Q > x) \approx e^{\frac{-(x-1-\kappa_{t_x})^2}{2\nu_{t_x}}} - e^{\frac{-(x-\kappa_{t_x})^2}{2\nu_{t_x}}} \qquad \textbf{Equation 3-18}$$

It is noted that the above equation is reached by assuming the invariant dominant time scale at queue length x-1 and x.

Formula 3.18 provides an estimate of the queue length distribution from the empirical traffic measurement. The main drawback of this scheme is its computational complexity of searching the dominant time scale for each queue length x. Additionally, the storage requirement is also a problem since a large data set is needed for searching for the dominant time scale.

Similar to the Large Deviation Theory, the Maximum Variance Theory provides an asymptotic solution for the queue length distribution, a tighter estimate is not guaranteed.

3.5 Summary

In this section, we account for the cause of the end-to-end delay in the network domain. The end-to-end delay consists of deterministic and stochastic components. The deterministic component is fixed for various data packet using the same path, (e.g. the link propagation delay), whereas, the stochastic component differs from packet to packet. The stochastic delay component includes the queuing delay, transmission delay, processing delay. We argue that the queuing delay is more significant than the other stochastic components. Therefore, the delay distribution excluding the deterministic part should be in accordance with the queue length distribution.

Currently, end-to-end delay measurement mainly relies on Active Measurement, e.g. Estimation by RTT or OWMP. Apart from the intrinsic weakness in Active Measurement, each measurement scheme has its own shortcomings. While the measurement result by RTT is very coarse, the clock-synchronization between two remote ends in OWMP is still a problem. Therefore, it is interesting to study the feasibility of employment of Passive Measurement for end-to-end measurement.

In our scheme which we will propose in Chapter 4, the queue is passively monitored and so no intrusion effect is created. Queue length monitoring is also found in various works. We present the related works and their operational principles.

Chapter 4 Introduction to QL Measurement

In this chapter, we present our measurement methodology, Queue Length (QL) Measurement. As its name suggests, our scheme is based on queue length monitoring. We will explain how the queue length is monitored and how to exploit measurement results to achieve our goals: (a) per-hop queue length distribution estimation and (b) end-to-end delay distribution estimation.

4.1 Queuing Behaviour

Similar to the work [KES99], our approach exploits prior knowledge of the queuing characteristic in the queue length measurements. As mentioned in Chapter 1, a switch/router is normally tailored for the fast packet forwarding/routing, it is unwise to add extra measurement burden on it. By using our queuing model, it enables our measurement technique to be implemented with low computational complexity. It also alleviates the storage requirement for the measurement result. The measurement result is used to infer the queue length distribution, while the normal network operation will not be much interfered with. The queuing behaviour is affected by not only the queuing discipline, but also the traffic characteristics. Traffic models have been long used to mimic the real network traffic. We highlight some most useful traffic models and their effects on the queuing behaviour in the following sections.

4.1.1 Traffic Models

The purpose of a traffic model is to mimic real network traffic. An accurate traffic model can assist the network performance analysis either through analytical techniques or computer simulations. In this thesis, our work was studied and examined through computer simulations using simulators OPNET and Network Simulator ns2 with different traffic models. A good traffic model should be able to capture the statistical characteristics of actual traffic. Among the numerous traffic models available, we employ ON/OFF traffic models for our simulation. In this section, we review some of the more well-recognized traffic models.

Traffic models can be stationary or non-stationary. For the stationary traffic model, it can be further divided into two main classes: short-range dependent (SRD) and long-range dependent (LRD). The short-range dependent traffic models have a correlation structure that is significant for relatively

small lags. On the contrary, long-range dependent traffic models have significant correlations even for large lags.

4.1.1.1 Short-Range Dependent Traffic Models

The first performance models of telecommunication systems were based on the assumption that arrival processes follow a Poisson process. It became, however, quickly apparent that this characterization ignores the significant correlation present in network traffic. For this reason, Markov modulated processes have been introduced for representing a variety of traffic models. In this section, we present this traffic model and its variants.

A. Markov and Embedded Markov Models

In Markov Models, there is a given state space $S = \{s_1, s_2, \ldots, s_M\}$. Let X_n be a random variable which defines the state at time n. The set of random variables $\{X_n\}$ will form a discrete Markov Chain. The Markov property is such that the probability of the next value $X_{n+1} = s_j$ depends only on the current state. In other words, Markov properties imply that the future depends on the current state and not on previous states nor on the time already spent in the current state. Therefore, this property restricts the random variables, which describes the time spent in a state to a geometric distribution in the discrete case and to an exponential distribution in the continuous case. A semi-Markov process is obtained by allowing the time between state transitions to follow an arbitrary probability distribution. If the time distribution between transitions is ignored, the sequence of states visited by the semi-Markov process will be a discrete time Markov Chain, and is referred to an embedded Markov Chain. Some of the variant Markov traffic models were introduced as follows.

A1. ON/OFF Models

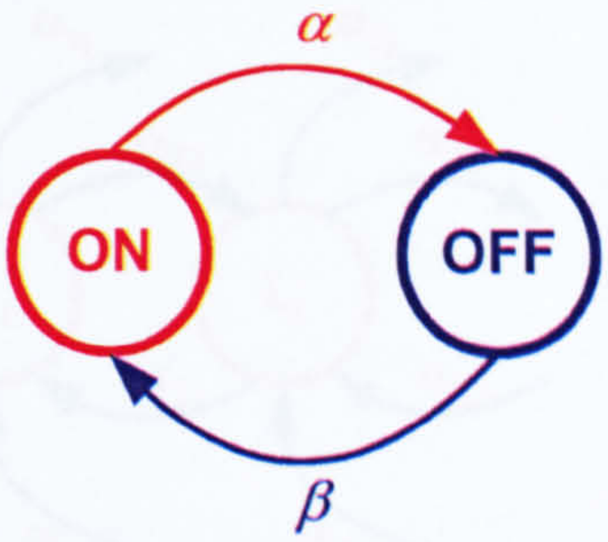


Figure 4-1 ON/OFF model

Figure 4-1 depicts an ON/OFF source model. This model is widely used for voice traffic modelling [HEF86] [NIK92]. These ON/OFF models have only two states such as ON state and OFF state as shown in Figure 4-1. Packets are only generated during talk spurts (ON state) with fixed inter-arrival time. The time spent in ON and OFF states is exponentially distributed with mean $1/\alpha$ and $1/\beta$ respectively and the Markovian property is maintained.

A2. Interrupted Poisson Process IPP Models

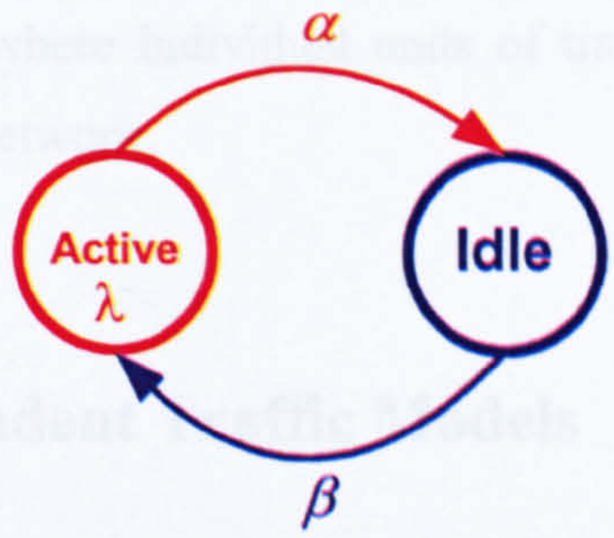


Figure 4-2 IPP model

Figure 4-2 depicts an IPP traffic model. Similar to the ON/OFF traffic model, there are two states in an IPP traffic model: Active and Idle. The time spent in Active and Idle states is exponentially distributed with mean $1/\alpha$ and $1/\beta$ respectively. Arrivals only occur in the active state, and do so according to a Poisson distribution with rate λ. Therefore, IPP and ON/OFF models differ only in inter-arrival time during the active (ON) state.

A3. Markov modulated Poisson process (MMPP)

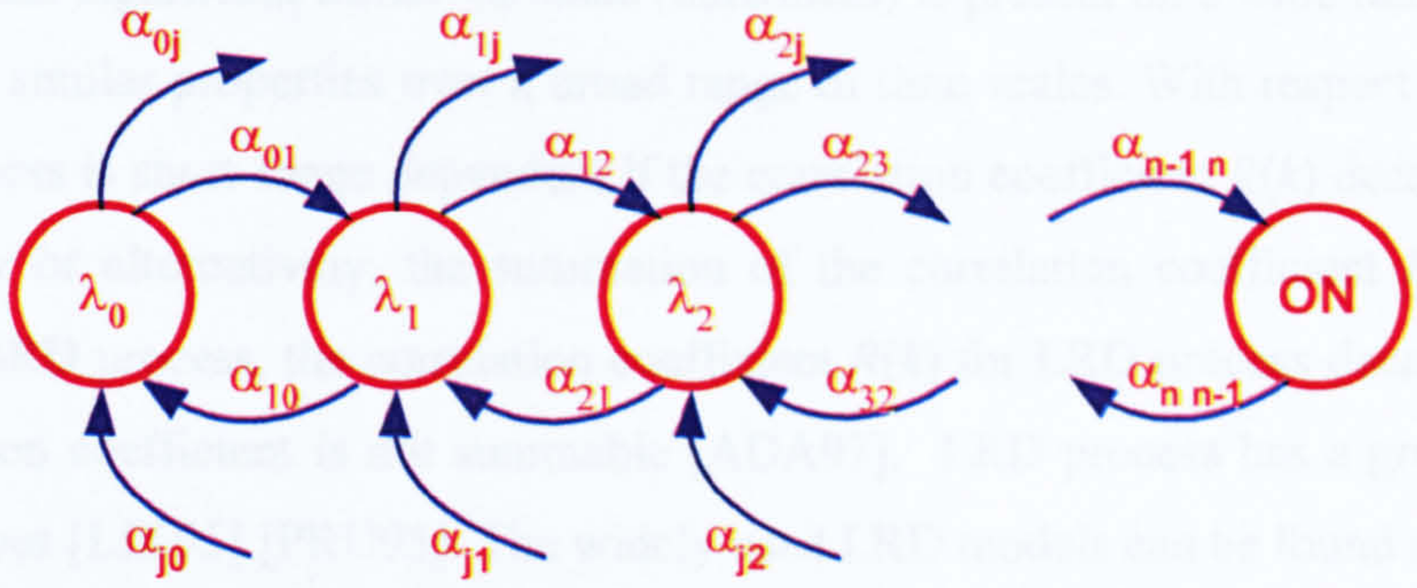


Figure 4-3 MMPP model

Figure 4-3 depicts an MMPP model. This model is also called a doubly stochastic process, and uses an auxiliary Markov process in which the current state of this Markov process controls the probability distribution of the arriving traffic. This model is fully characterized by the transition matrix $\{\alpha_{ij}\}$ and the Poisson arrival rate in each state λ_i. An MMPP process with M+1 states can be obtained by the superposition of M identical independent IPP sources. This process is extensively used to model aggregate network traffic. The variants of MMPP can be Markov-Modulated Fluid Flow and Markov-Modulated Bernoulli Process MMBP. For MMBP, time is discretized into fixed-length slots and the process spends a geometric duration of time slots in each state. The Markov-Modulated Fluid Flow model generates data at a constant rate when the system is in state *i*. This model is appropriate in the case where individual units of traffic i.e. packets or cells have little impact on the performance of the network.

4.1.1.2 Long-Range Dependent Traffic Models

The models described in the previous section have a correlation structure that is characterized by an exponential decay. However, a recent analysis of traffic measurements of Ethernet LAN traffic [LEL94] and NSFNET [KLI94], Web traffic [PAX95][CRO95] and ATM-encoded variable bit rate video [GAR94] has suggested that the auto-correlation function decays to zero at a slower rate than exponential. Taking Web traffic as an example, the source of the long-range dependent traffic comes from the heavy-tailed web transmission time which is primarily due to the heavy-tailed distribution of available file sizes in the Web [CRO95]. In addition, the influence of user "think time" also contributes to the long-range dependence web traffic property [CRO95]. If the traffic follows a Poisson or Markovian arrival process, it would have a characteristic burst length which would tend

to be smoothed by averaging over a long enough time scale whereas measurements of some real traffic indicate that significant traffic variance (burstiness) is present on a wide range of time scales. Such traffic has similar properties over a broad range of time scales. With respect to the correlation structure, a process is short-range dependent if the correlation coefficient $R(k)$ decays at least as fast as exponentially or alternatively, the summation of the correlation coefficient $R(k)$ is finite. In contrast to the SRD process, the correlation coefficient $R(k)$ for LRD process decays hyperbolically and its correlation coefficient is not summable [ADA97]. LRD process has a great impact on the queuing behaviour [LIK95] [PRU95]. The widely used LRD models can be found as follows:

A.1 Superposition of High Variability ON/OFF Sources

The ON/OFF source models discussed in Section 4.1.1.1 assume finite variance (exponential) distributions for the sojourn time in ON and OFF periods. Therefore, the aggregation of large number of such sources will not have significant correlation, except possibly in the short range.

An extension to such traditional ON/OFF models was first introduced by [TAQ86] by allowing the ON and OFF periods to have infinite variance (high variability or Noah Effect). The superposition of many such sources produces aggregate traffic that exhibits long-range dependence.

Pareto distribution is a natural choice for the heavy-tailed (possibly infinite variance) distribution. A Pareto distribution has the following probability density function [KRA02]:

$$F(x) = \frac{\alpha b^{\alpha}}{x^{\alpha+1}} \quad , \quad x \geq b \qquad \textbf{Equation 4-1}$$

where α is the shape parameter (tail index), and b known as location parameter is minimum value of x. When $\alpha \leq 2$, the variance of the distribution is infinite. When $\alpha \leq 1$, the mean value is infinite as well. For self-similar traffic, α should be between 1 and 2. The lower the value of α, the higher the probability of an extremely large x.

The Hurst parameter, H, is an effective index to show the degree of the burstiness of the traffic train [ADA97]. For the SRD, the Hurst parameter is equal to 0.5, whereas, the Hurst parameter is between 0.5 and 1 for the LRD traffic. The higher the value of Hurst parameter H is, the stronger the long range dependence.

To generate LRD traffic with a superposition of high variability ON/OFF sources, the Hurst parameter H is given as follows [TAQ97].

$$H \approx \frac{3-\alpha_{\min}}{2} \qquad \textbf{Equation 4-2}$$

where $\alpha_{\min}$ is the minimum value of the shape parameters defined for ON and OFF periods.

A.2 Fractional Brownian Motion

Fractional Brownian Motion is a modified version of Brownian Motion. Brownian Motion is also known as a Wiener Process. A random process $\{B\{t\}, t \geq 0\}$ is called a Wiener Process if it possesses the following properties:

- $B(t)$ has stationary independent increments
- The increment $B(t)$-$B(s)$ ($t>s$) is Normally distributed
- $E[B(t)] = 0$
- $B(0) = 0$

It can be shown that the variance of a Brownian Motion Process is equal to $\sigma^2 t$, where σ^2 is a parameter of the Wiener process which must be determined from observation. The autocorrelation function $R_B(t,s) = \sigma^2 \min(t,s)$ [HSU97]. Fractional Brownian Motion differs from Brownian Motion by having increments with variance $\sigma^2 t^{2H}$. Under this condition, Fractional Brownian Motion $fB(t)$ has an autocorrelation function as follows:

$$R_{fB}(k) = \frac{1}{2}\left([k+1]^{2H} - 2k^{2H} + [k-1]^{2H}\right) \quad , k \geq 0 \qquad \textbf{Equation 4-3}$$

It can be noted that Equation 4-3 satisfies the long-range dependence criteria as the correlation function becomes Equation 4-4 by taking Taylor series.

$$R_{fB}(k) \approx H(2H-1)k^{2H-2} \quad , k \to \infty \qquad \textbf{Equation 4-4}$$

Therefore, the Fractional Brownian Motion process exhibits long-range dependence when H is between 0.5 and 1 and the autocorrelation decays hyperbolically. [MAN69] [KOG95] discussed how to generate Fractional Brownian Motion processes.

4.1.2 Why ON/OFF models?

In this thesis, a superposition of ON/OFF sources is chosen to generate both the SRD and LRD traffic respectively. To generate SRD traffic, the ON and OFF periods are governed by the exponential distribution, whereas, a heavy-tailed distribution Pareto distribution is used for LRD traffic. The advantages of using ON/OFF models are:

- Simplicity: few parameters are required for defining the models
- A basic prototype for the bursty traffic source
- An effective model used for mimicking packet traffic

To model the traffic by using aggregate ON/OFF source, only a few parameters are needed compared with other traffic models like MMPP. The parameters include the number of the aggregate sources, the packet transmission rate in the ON period and the parameters defined in the distributions for the ON and OFF periods. If exponential distributions are employed for the ON/OFF periods, then the mean ON time and mean OFF time are sufficient. When a Pareto distribution is used, in addition to the mean ON and OFF time, the shape parameter is also required.

ON/OFF behaviour is the basic nature of packet traffic. For example, with voice traffic, packets are generated when the user is in spurt state, whereas, in the silent state, a small amount of packets or no packets are generated and the process alternates between ON and OFF state. Another obvious example of the ON/OFF behaviour can be found in Web traffic. Users download the web-page information which is equivalent to the ON state, and then it switches to OFF state for browsing this web-page. An ON/OFF source is the basic prototype for a bursty source and has been used extensively in traffic modelling [MIC97].

Superposition of multiple ON/OFF sources has been found to effectively mimic real traffic [BHA00]. The use of superposition of ON/OFF sources is found frequently in many sources of literature. [BAI91] [HAN94] uses superposition of ON/OFF models in their work.

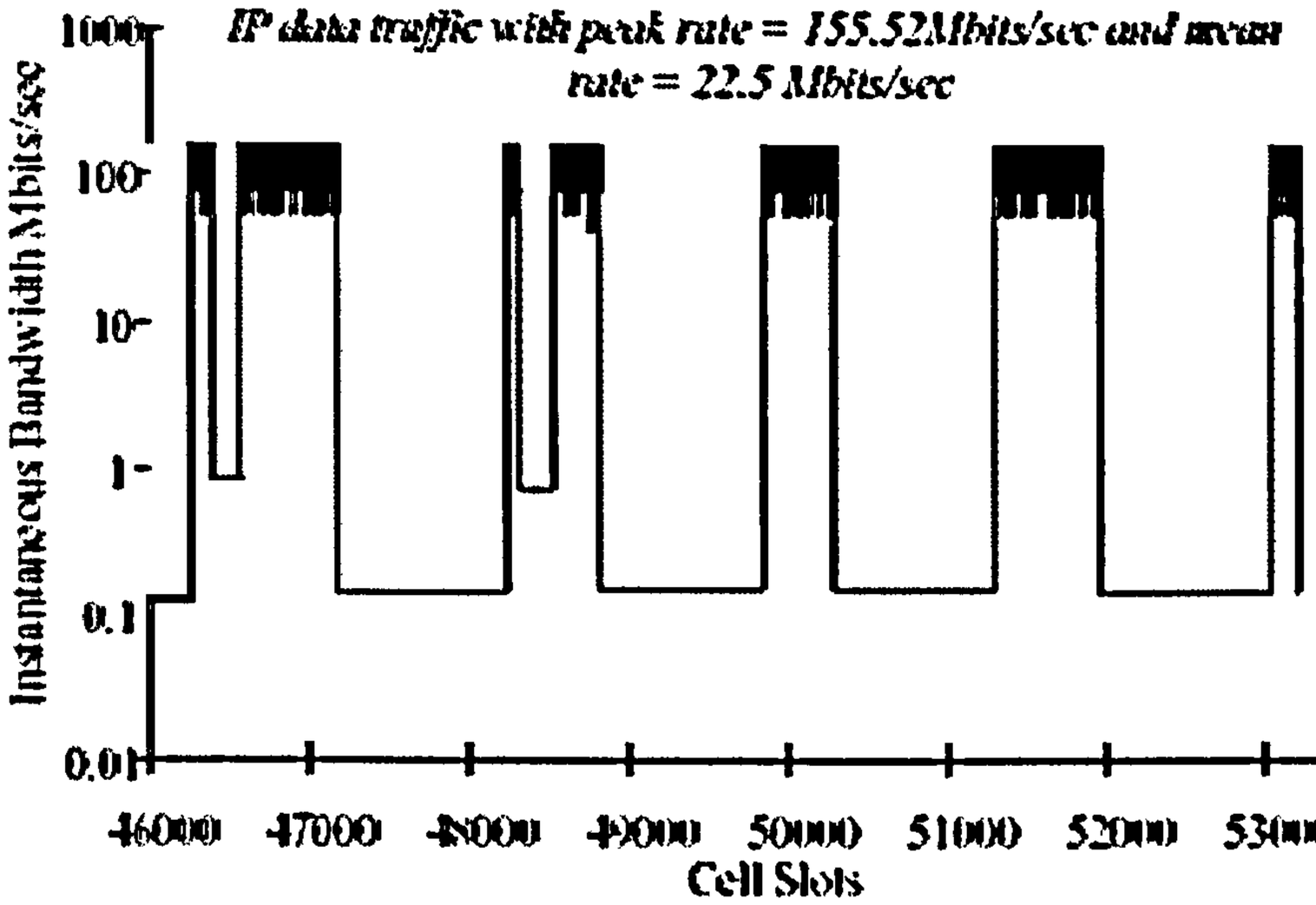


Figure 4-4 IP traffic trace [HOL00]

Various measurements taken in data networks reveal the ON/OFF traffic nature at the aggregate level [GAR00]. The trace in Figure 4-4 represents IP traffic obtained from simulations run on a real network with live traffic as part of the European ACTS Expert Project [EXP94]. With reference to this Figure, it can be noted that the real aggregate traffic in IP networks exhibit ON/OFF switching behaviour. It is believed that ON/OFF traffic sources is effective to mimic data network traffic. For the sake of brevity, in this thesis, the traffic trace produced by the superposition of exponential ON/OFF sources is called *Markovian traffic*, whereas, the traffic trace produced by the superposition of high variability ON/OFF sources is called *Power-law traffic*.

4.1.3 Queue Length Distribution for Markovian Traffic

A variety of queuing analysis was performed to study the queue length behaviour under various conditions in the past decades. One of the approaches is known as Large Deviation Theory. This method attracts much interest on the studying the cell-loss probability in the broadband network. As discussed in Section 3.4.1.2.A, LDT reveals the fact that the queue length distribution decays exponentially for a FIFO queue multiplexing a large class of arrival process which includes Markovian traffic. The exponential decay of the queue tail distribution has also been studied in various papers [ABA95] [JEL95] [PIT00]. In [ABA95], it gives the analysis for the exponential queue tail distribution for the traffic model MMPP and BMAP (Batch Markovian Arrival Process).

Excess Rate (ER) Analysis in [PIT00] was used to analyse the rate of exponential decay of the queue tail distribution.

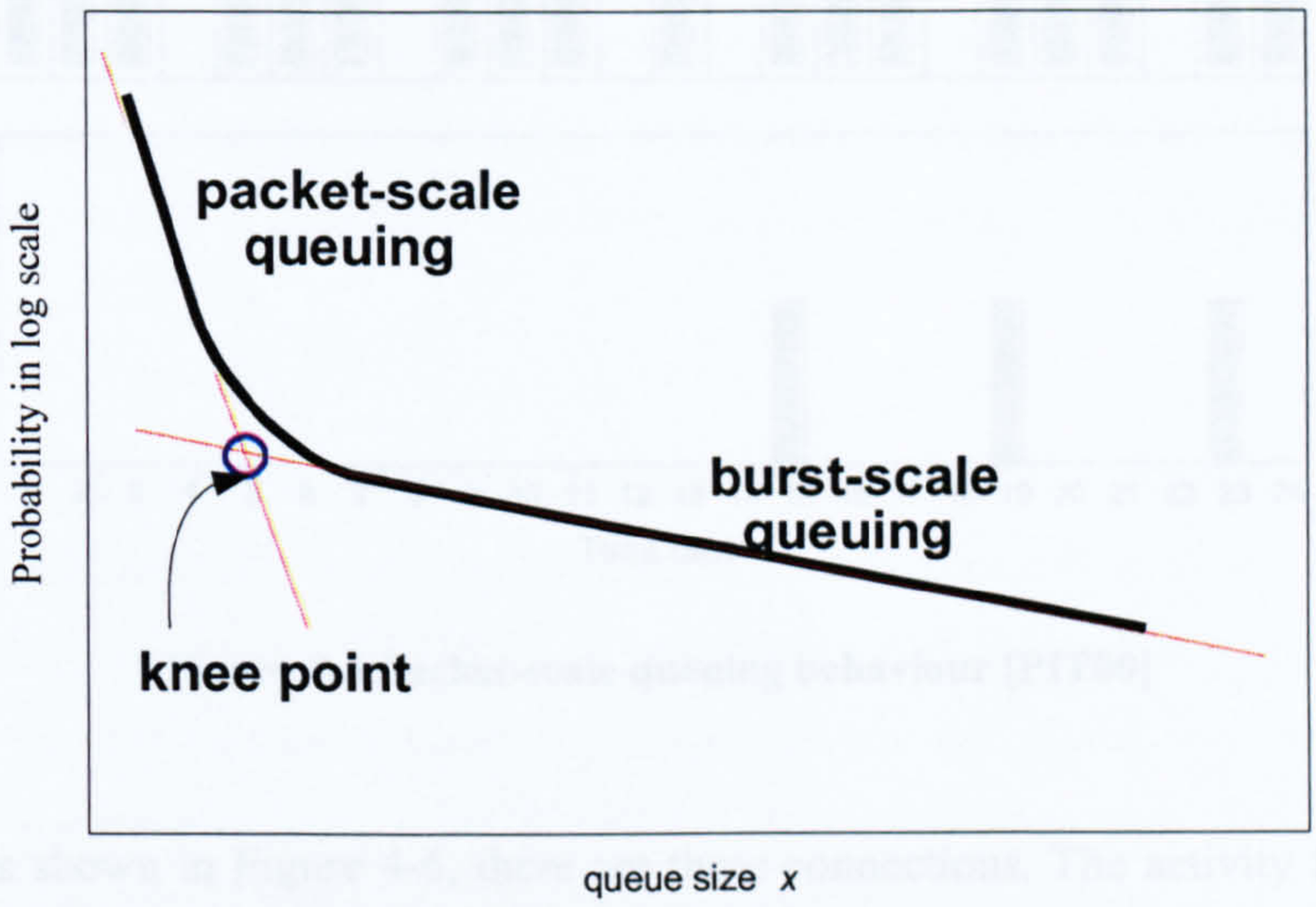


Figure 4-5 Queue length distribution for a FIFO multiplexing Markovian traffic

Figure 4-5 depicts the typical queue length distribution for a FIFO queue multiplexing Markovian traffic. With reference to Figure 4-5, the queue length distribution consists of two regions, often referred to as packet-scale region and burst-scale region [PIT00]. The queue length distribution starts with the packet component with a steeper slope. The distribution then gradually changes to a second branch with a smaller slope, the burst component. The intersection of the tangents to both branches is often designated as the "knee" point. The qualitative explanation of this behaviour is given in the following sections.

4.1.3.1 Packet-Scale Queuing

To account for the packet-scale queuing, it is easier to view the aggregate traffic produced by a group of individual connections. Each connection switches between ON and OFF mode. Packets are generated in ON state. When the number of connections in ON states is relatively small, the equivalent packet arrival rate is smaller than the queue service rate. Then, the queue is most likely to be empty. Nevertheless, a smaller queue may happen due to the random arrival of the packets. This phenomenon is illustrated in Figure 4-6.

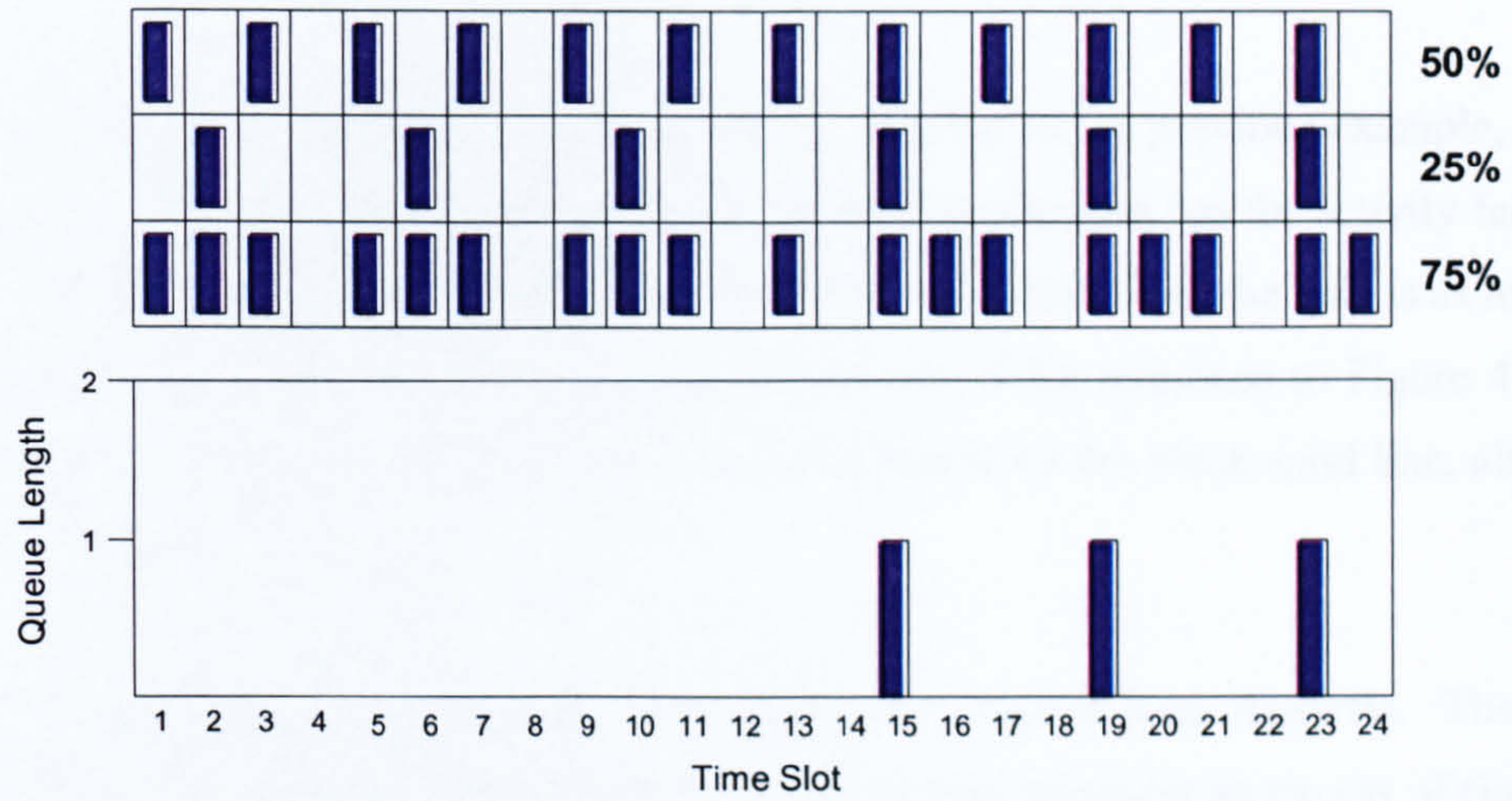


Figure 4-6 Packet-scale queuing behaviour [PIT00]

In the example as shown in Figure 4-6, there are three connections. The activity factor of the first, second and third connection are 0.5, 0.25 and 0.75 respectively. Two packets can be removed from the queue during a time slot. Under this configuration, the load is equal to 0.75. From the diagram, it can be noted that only a short queue exists. In other words, only packet-scale queuing behaviour occurs. The small queue is established owing to the random arrival pattern of the incoming packets.

4.1.3.2 Burst-Scale Queuing

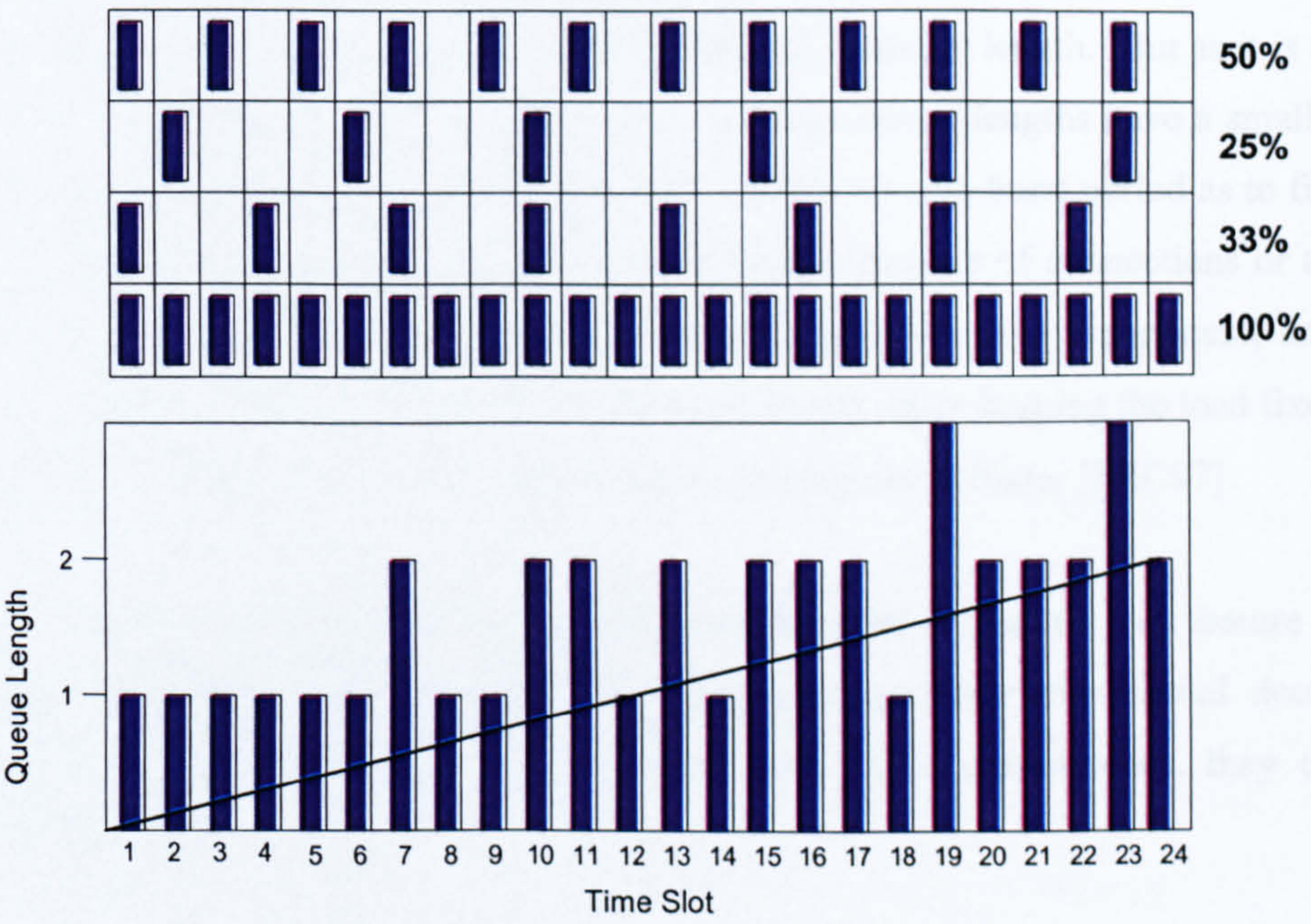


Figure 4-7 Burst-scale queuing behaviour [PIT00]

Figure 4-7 illustrates the burst-scale queuing effect. Similar to the previous example, two packets can be served in each time slot. In this example, the third connection has the activity factor equal to 33% and an extra connection is added with an unity activity factor, then, the load is as high as 1.041. The equivalent arrival rate is greater than the service rate. With reference to Figure 4-7, it can be noted that the queue length is gradually built up as indicated by the black solid line, although some fluctuation is shown.

The rate of increasing queue length can be analysed with Excess Rate Analysis. The Excess-rate (ER) packets refer to "packets which must be queued as they represent an excess of (instantaneous) arrival rate over the service rate" [PIT96]. These Excess-rate packets are important in the ER analysis. Taking Figure 4-7 as an example, on average, the number of packets generated in each time slot by these four connections is equal to 0.5+0.25+0.33+1 = 2.0833 packets. As a result, the number of ER packets is equal to 0.0833 packets per time slot as the service rate is equal to 2 packets per time slot. With the rate of increasing queue length = 0.0833 packets per time slot, it will take 24 time slots to reach the queue length = 2 packets as shown in Figure 4-7.

As discussed above, the packet-scale queuing behaviour is caused by the accidental coincidence of packet arrivals of two or more connections when the simultaneous transmission rate is smaller than the service rate. When several connections emit bursts of packets simultaneously, and the transmission rate is greater than the service rate, when it lasts for a considerable longer period than a time-unit, the queue begins to fill and the burst-scale queuing effect happens. The more connections are in a burst period simultaneously, the larger the growth in queue length. But as it is unlikely that all connections send bursts at the same time, the very large queue lengths have a small probability. The knee position, in general, is where just enough sources are in a burst period as to fill the queue. There are some general trends in this behaviour. When the number of connections or the peak rate increases, the knee moves upward, reducing the packet-scale queuing component, and the burst-scale component becomes flatter. Increasing the burst length while keeping the load fixed leaves the knee point more or less unchanged but makes the burst component flatter [MIC97].

For a FIFO queue multiplexing Markovian traffic, as discussed before, the key feature of the burst-scale queuing and the packet-scale queuing components are their exponential decay, i.e. they approximately follow a straight line in log-linear scale. As a consequence, they can be well-modelled as follows:

- Packet-scale queuing component

$$Q_m(x) = c_{pm}\eta_{pm}{}^{x} \qquad \textbf{Equation 4-5}$$

- Burst-scale queuing component

$$Q_m(x) = c_{bm}\eta_{bm}{}^{x} \qquad \textbf{Equation 4-6}$$

where $Q_m(x)$ denotes the queue length distribution. The parameters c_{pm} and c_{bm} denote packet-scale and burst-scale decay constant, η_{pm} and η_{bm} denote packet-scale and burst-scale decay rate respectively. The logarithms of the decay constants represent the offset of the distribution in the log-linear plot, whereas the logarithms of the decay rates represent the slope of the distribution.

4.1.4 Queue Length Distribution for Power-law Traffic

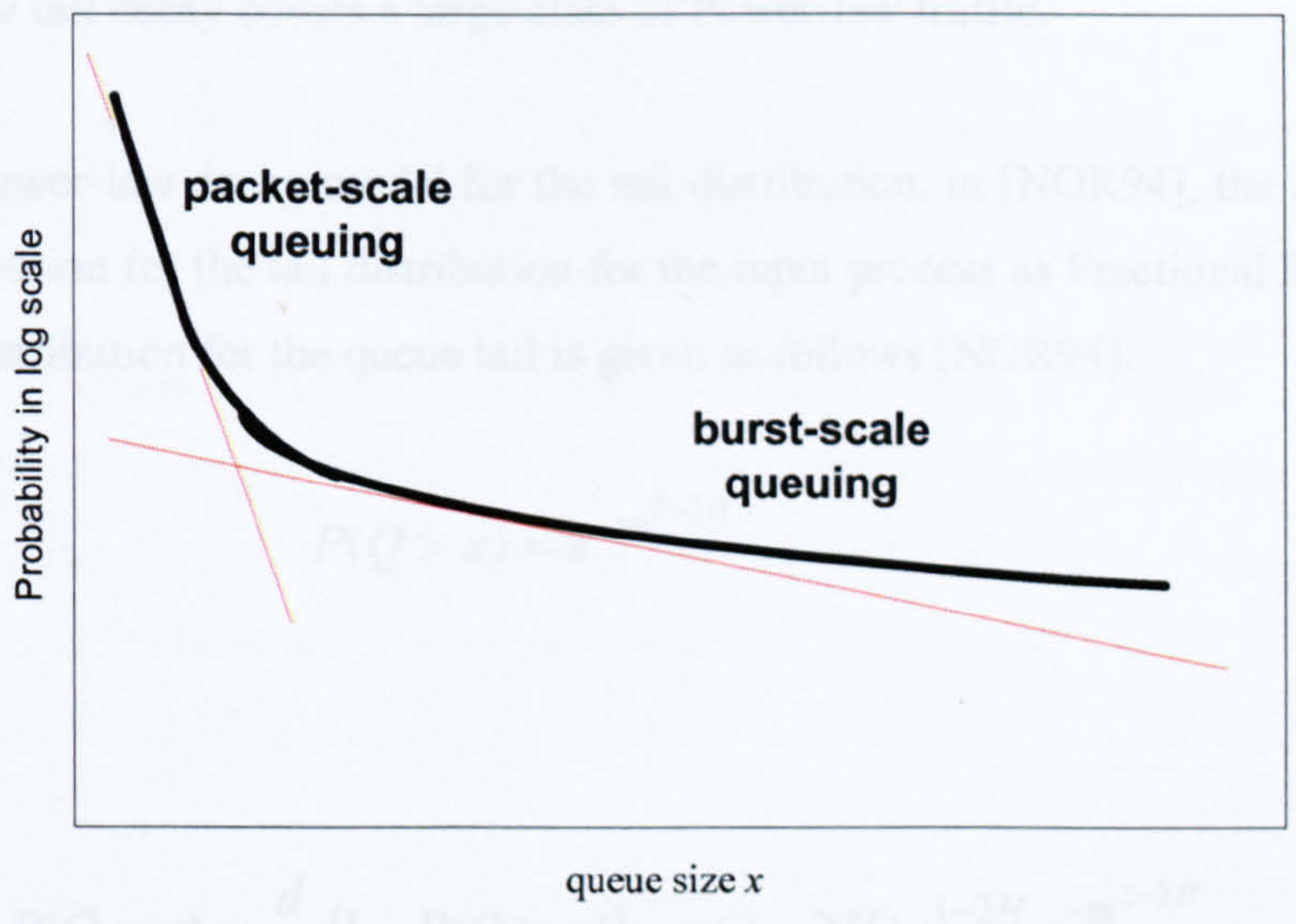


Figure 4-8 Queue length distribution in the presence of Power-law traffic

Figure 4-8 depicts the typical queue length distribution for a FIFO multiplexing Power-law traffic. Similar to the distribution of Markovian traffic, the queue length distribution still combines two components such as packet-scale region and burst-scale region. The cause of packet-scale queuing behaviour is the same as that in the Markovian case, i.e a small queue occurs due to the random arrivals of packets, but the simultaneously transmission rate is lower than the service rate. Regarding the burst-scale queuing region, the shape of queue tail distribution in the presence of Power-law traffic differs from that of Markovian traffic. As discussed in the previous section, we illustrated

that the queue tail for a FIFO queue multiplexing Markovian traffic has an exponential decay. However, in the case of Power-law traffic, the decay rate in the tail will not be as fast as exponential.

In contrast to Markovian traffic, the continuous bursty nature of Power-law traffic in different time-scales causes different queuing behaviour. Heavy-tailed distribution such as Weibullian distribution and power-law decay distribution are reported to model the queue tail distribution in the presence of Power-law traffic.

In the work [LIK95] [PRU95] [PRU95a] [NOR96] [MA00] [MA02], the authors gave the mathematical proof of the power-law decay behaviour in the queue tail distribution in the presence of Power-law traffic. In [PRU95a], the proof is given under the condition that the queue is driven by a single, infinite variance ON/OFF source, whereas power-law tail distribution decay with multiple infinite variance ON/OFF sources was studied in [MA02]. Furthermore, the queue driven by traffic generated by Chaotic Map, another proposed effective Power-law traffic generation algorithm, also demonstrates the power-law decay in the queue tail distribution [PRU95]. Therefore, it is expected that the power-law tail decay covers a large class of Power-law traffic.

Apart from the power-law decay model for the tail distribution, in [NOR94], the authors report the Webuillian distribution for the tail distribution for the input process as Fractional Brownian Motion. The Webullian distribution for the queue tail is given as follows [NOR94]:

$$P(Q > x) \approx e^{-\gamma x^{2-2H}} \qquad \textbf{Equation 4-7}$$

Alternatively,

$$Q(x) = P(Q = x) = \frac{d}{dx}\left(1 - P(Q > x)\right) \approx \gamma(2-2H)x^{1-2H}e^{-\gamma x^{2-2H}} \qquad \textbf{Equation 4-8}$$

where H is the Hurst parameter of the Fractional Brownian Motion and the coefficient γ is a function of the mean arrival rate and Hurst parameter of the Fractional Brownian Motion [NOR94]. With reference to Equation 4-8, it can be noted that the expression consists of two main factors as x^{1-2H} and $e^{-\gamma x^{2-2H}}$. The first factor is the power-law decay component, whereas, the latter is the modifier.

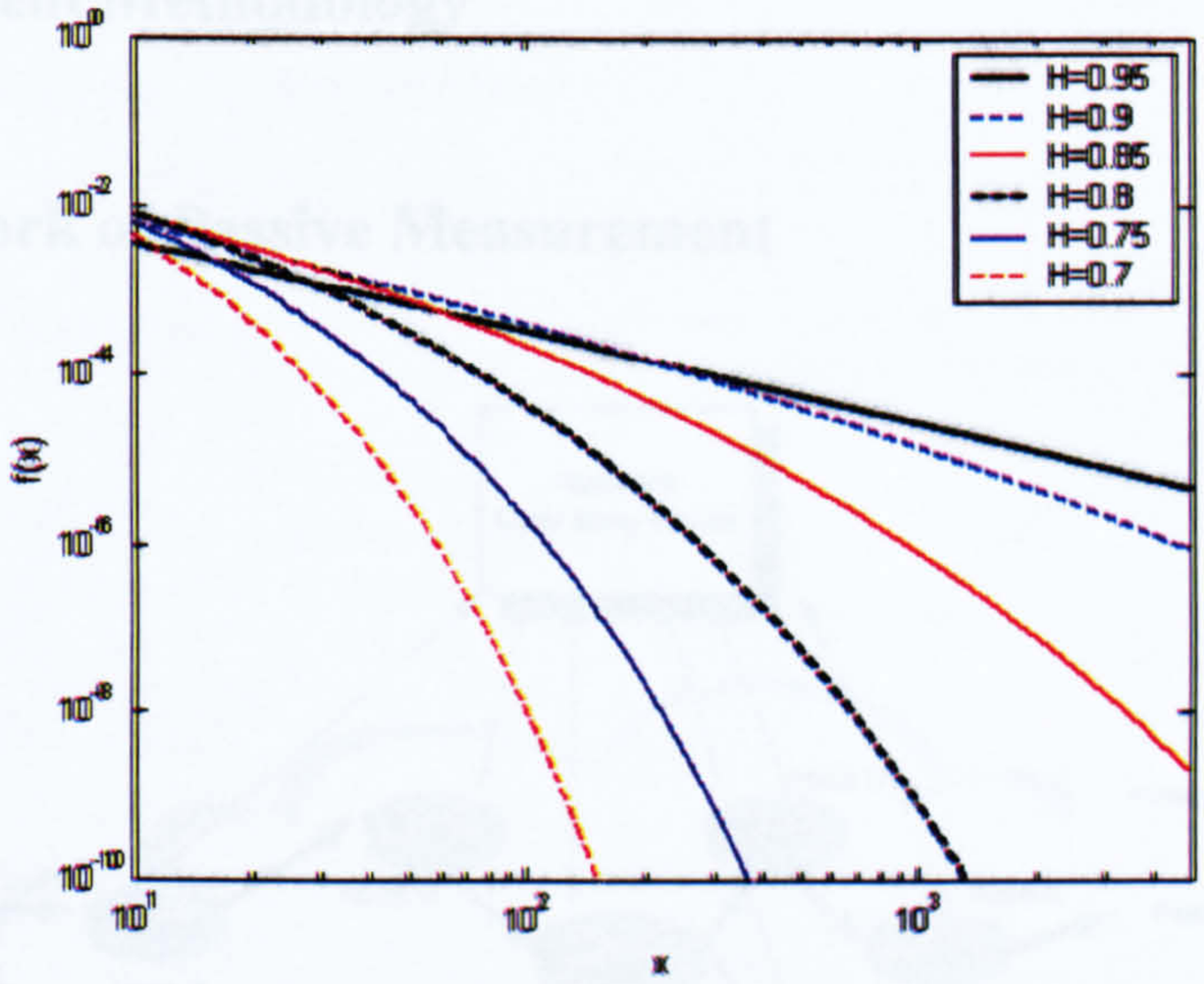


Figure 4-9 Plot of the standard Weibullian distribution

Figure 4-9 depicts the plot of the standard Weibullian distribution with various Hurst parameters. It can be noted that the power-law decay component is dominant for a larger value of Hurst parameter and so the distribution exhibits a straight line in a log-log scale. On the other hand, there is a curvature in the tail for a smaller value of Hurst parameter such as H=0.7.

The power-law tail decay model covers a large class of Power-law traffic models. But the Wellbullian distribution is also reported. If its tail is approximated by a straight line in a log-log plot, then the queue tail distribution or the burst-scale queuing region also follows the power-law decay model. Therefore, the power-law decay tail model is the natural choice for a FIFO queue multiplexing Power-law traffic. The model is shown as follows:

$$Q_p(x) = c_{bp} x^{\eta_{bp}} \qquad \textbf{Equation 4-9}$$

where c_{bp} and η_{bp} denote the burst-scale decay constant and burst-scale decay rate respectively. The logarithm of the burst-scale decay represents the offset, whereas the burst-scale decay rate represents the slope of the queue tail distribution in a log-log plot.

4.2 Measurement Methodology

4.2.1 Framework of Passive Measurement

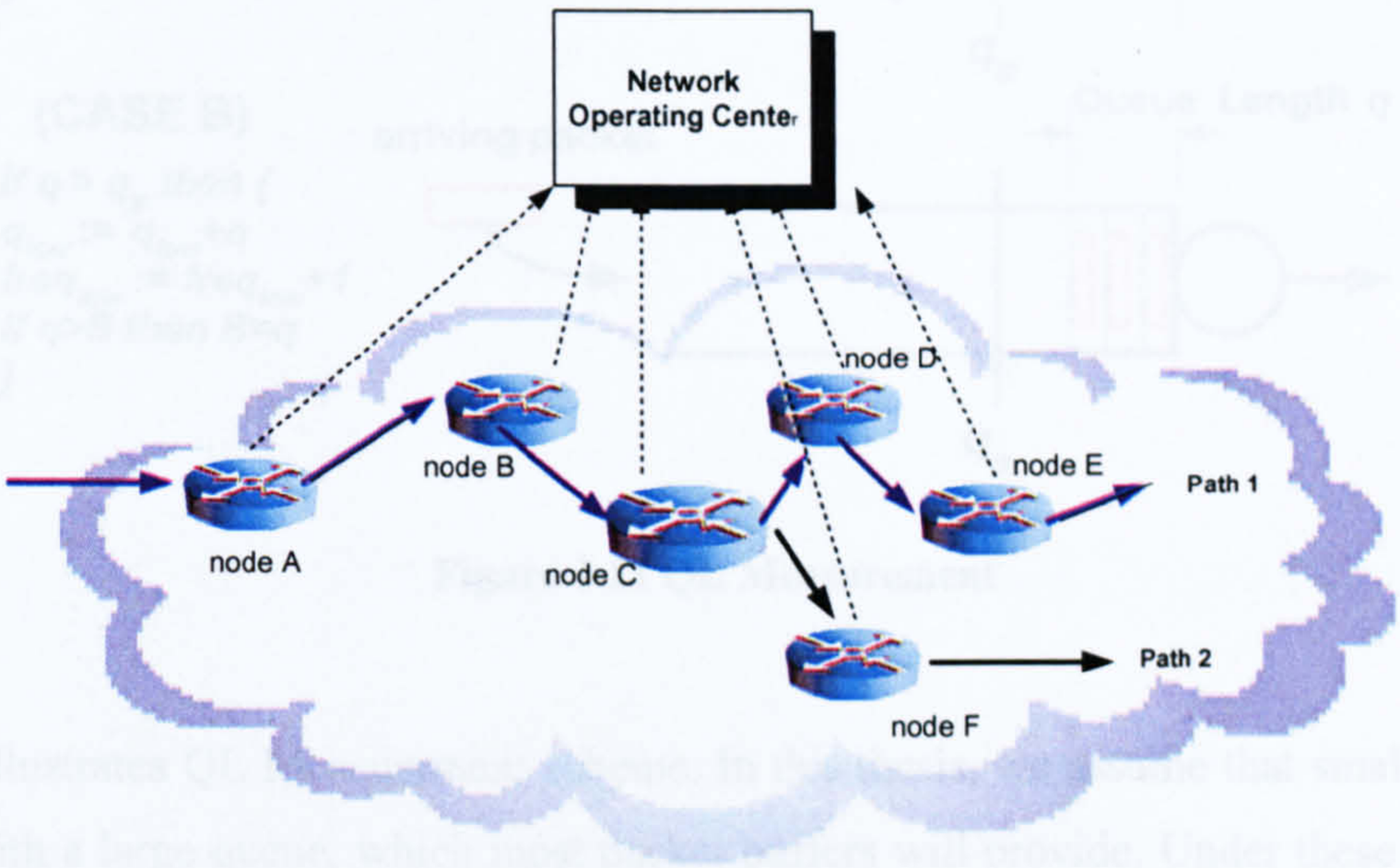


Figure 4-10 The framework of distributed measurement

Figure 4-10 illustrates the framework of our measurement scheme. The measurement process is distributed in every node along the measuring path. The passive queue length measurement "QL" Measurement is performed in each node. The measurement results are collected by the NOC via the management system e.g. polling MIB as discussed in Section 2.3.2.C. The NOC is responsible for re-constructing the queue length distribution at each network node. Then, the end-to-end delay distribution can be determined by using convolution on every local delay distribution of every node along the path [VER99] [SEU00] [ØST02] as discussed in Section 3.3.C.

4.2.2 QL Measurement – at Local Node

Accurate queue length distribution estimation is the key to our success. In this section, we will explain how it works.

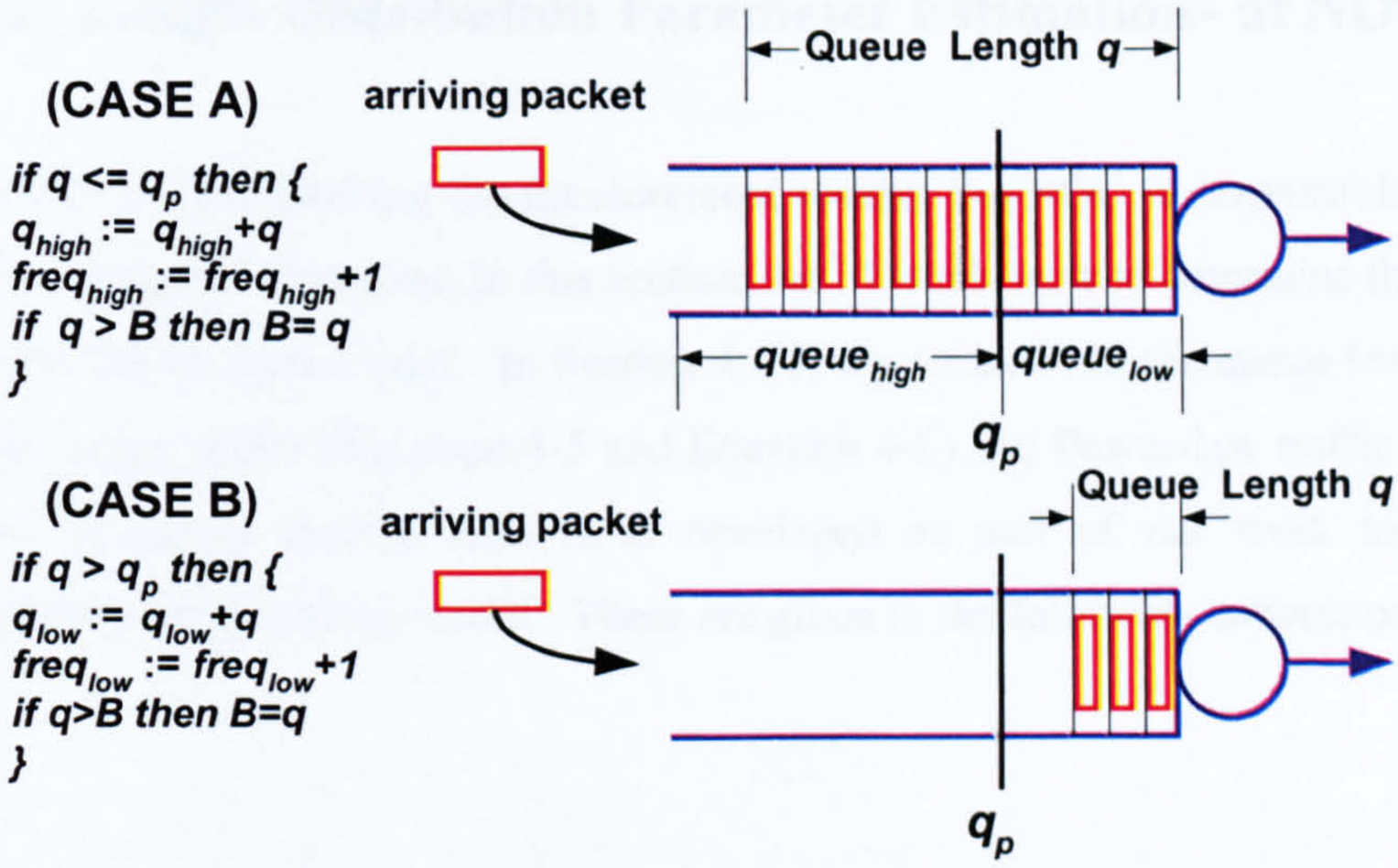


Figure 4-11 QL Measurement

Figure 4-11 illustrates QL Measurement scheme. In this thesis, we assume that small packet loss is guaranteed with a large queue, which most packet buffers will provide. Under these circumstances, delay performance is more relevant. With reference to Figure 4-11, the queue is logically partitioned into two regions such as $queue_{high}$ and $queue_{low}$ by a partition point called q_p. The purpose of this partition point is to isolate the packet-scale region from the burst-scale region for the measurement. The partition point should be large enough to be placed in the burst-scale region[6].

Five measurement data are recorded during the measurement period. When the measurement starts all the measurement data q_{high}, $freq_{high}$, q_{low}, $freq_{low}$ and B are initialised to zero. If the current queue length seen by an incoming packet is greater than the partition point, then the data q_{high} and $freq_{high}$ are updated, otherwise q_{low} and $freq_{low}$ as shown in Figure 4-11. When the present measurement period finishes, the measurement data q_{high}, $freq_{high}$, q_{low}, $freq_{low}$ and B will be stored for the queue length distribution construction. The measurement results $freq_{low}$ and $freq_{high}$ represents the frequency of the $queue_{low}$ and $queue_{high}$ region being hit.

It can be noted that the measurement scheme includes a few simple comparison and addition processes. These processes will not add much complexity on the local network node and the efficiency of the packet forwarding will be maintained.

---

[6] As we note later, this does not present a particular problem of accuracy as shown by the results in Section 5.3.1. This is a strength of our proposal.

4.2.3 Queue Length Distribution Parameter Estimation - at NOC

As discussed above, after fetching the measurement results, the NOC is responsible for generating each local queue length distribution. In this section, we illustrate how to determine the parameters of the queue length distribution model. In Section 4.1.3, we pointed out the queue length distribution model for Markovian traffic (Equation 4-5 and Equation 4-6) and Power-law traffic (Equation 4-9). Two different analytical models have been developed as part of our work to determine the parameters for the corresponding model. These are given in the following subsections.

4.2.3.1 Queue Length Distribution Re-construction (Markovian Traffic)

Recall that the measurement data consists of $queue_{low}$, $queue_{high}$, $freq_{low}$, $freq_{high}$ and B, $freq_{low}$ and $freq_{high}$ represents the frequency of the $queue_{low}$ and $queue_{high}$ being hit. Therefore, the probability of hitting these two regions can be estimated as follows:

$$p_{low} = \frac{freq_{low}}{freq_{low} + freq_{high}} \qquad \textbf{Equation 4-10}$$

$$p_{high} = \frac{freq_{high}}{freq_{low} + freq_{high}} \qquad \textbf{Equation 4-11}$$

where p_{low}, p_{high} are the estimate of the probability hitting $queue_{low}$ and $queue_{high}$ region respectively.

The conditional mean queue length in $queue_{low}$ region and $queue_{high}$ region are given by $\bar{q}_{low}$ and $\bar{q}_{high}$ respectively as follows:

$$\bar{q}_{low} = \frac{q_{low}}{freq_{low}} \qquad \textbf{Equation 4-12}$$

$$\bar{q}_{high} = \frac{q_{high}}{freq_{high}} \qquad \textbf{Equation 4-13}$$

Recall that the queue tail distribution decays exponentially for Markovian traffic as shown in Equation 4-6, by assuming the partition point is in the burst-scale region and the buffer size is large, then we can relate $\bar{q}_{high}$ with the queue tail distribution parameters: burst-scale decay rate η_{bm}, decay constant c_{bm} and the partition point q_p.

$$\bar{q}_{high} = \sum_{j=q_p+1}^{\infty} j \frac{c_{bm}\eta_{bm}{}^{j}}{\sum_{k=q_p+1}^{\infty} c_{bm}\eta_{bm}{}^{k}} \qquad \textbf{Equation 4-14}$$

With some series manipulations, we can obtain the following expression: (The proof is given in Appendix B)

$$\hat{\eta}_{bm} = 1 - \frac{1}{\bar{q}_{high} - q_p} \qquad \textbf{Equation 4-15}$$

$\bar{q}_{high}$ is determined by the measurement data as shown in Equation 4-13. Hence, the formula in Equation 4-15 can be used to estimate the burst-scale decay rate η_{bm}. This estimate $\hat{\eta}_{bm}$ is also a Maximum Likelihood Estimate [HSU97] for the burst-scale decay rate. The proof is given in Appendix C.

The parameter burst-scale constant c_{bm} can be determined by relating the measurement result p_{high} with the queue tail distribution model. Since p_{high} represents the probability of hitting $queue_{high}$ region, p_{high} can be expressed as follows:

$$p_{high} = \sum_{k=q_p+1}^{\infty} c_{bm}\eta_{bm}{}^{k} = c_{bm} \cdot \frac{\eta_{bm}{}^{q_p+1}}{1-\eta_{bm}} \qquad \textbf{Equation 4-16}$$

Rearranging Equation 4-16 and substituting the burst-scale decay rate estimate in Equation 4-15, then we obtain the estimate for the decay constant.

$$\hat{c}_{bm} = \frac{1-\hat{\eta}_{bm}}{\hat{\eta}_{bm}{}^{q_p+1}} p_{high} \qquad \textbf{Equation 4-17}$$

Now, this completes the estimation process for the tail of queue length distribution, i.e. the burst-scale region.

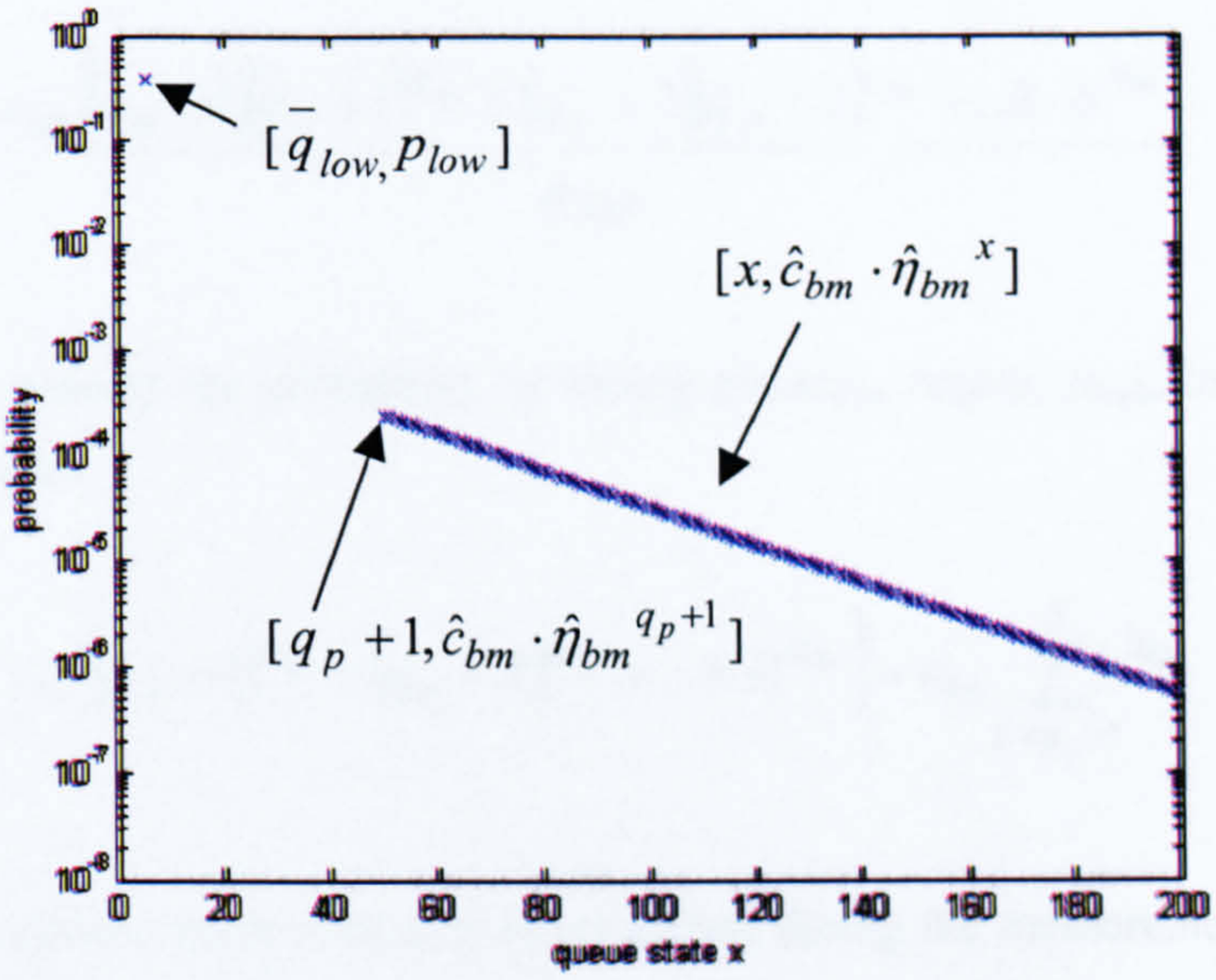


Figure 4-12 Re-construction of queue length distribution $\hat{Q}_m(x)$

Figure 4-12 depicts the queue length distribution re-construction. This can be described as follows:

- In $queue_{low}$ region, i.e. queue state $X \leq q_p$:

  $\hat{Q}_m(X = \bar{q}_{low}) = p_{low}$

  elsewhere, the queue state probability = 0

- In $queue_{high}$ region, i.e. queue state $X > q_p$:

  $\hat{Q}_m(X = x) = \hat{c}_{bm} \cdot \hat{\eta}_{bm}{}^{x}$

It is noted that a single point is used to represent the queue state probability in the $queue_{low}$ region, whereas, the $queue_{high}$ region is represented by an exponential decay curve with its decay constant and decay rate obtained from Equation 4-17 and Equation 4-15 respectively. Reference [PIT00] states that the packet-scale region is normally within the range of the order of a few 10's of packets and the packet-scale region has less impact on the end-to-end delay tail distribution as burst-scale region. This accounts for the reason why the $queue_{low}$ region is coarsely approximated by a single point.

4.2.3.2 Queue Length Distribution Re-construction (Power-law Traffic)

Similar to the approach in the case of Markovian traffic, we determine the estimate of the decay rate by relating the conditional mean queue length in $queue_{high}$ region, $\bar{q}_{high}$, with the queue tail model in Equation 4-9 as follows:

$$\bar{q}_{high} = \frac{c_{bp}\left\{(q_p+1)(q_p+1)^{\eta_{bp}} + (q_p+2)(q_p+2)^{\eta_{bp}} + ..B \cdot B^{\eta_{bp}}\right\}}{p_{high}} \quad \text{Equation 4-18}$$

Likewise, we can express the probability of hitting $queue_{high}$ region, p_{high}, in terms of queue tail distribution as follows:

$$p_{high} = c_{bp}\left\{(q_p+1)^{\eta_{bp}} + (q_p+2)^{\eta_{bp}} + ... + B^{\eta_{bp}}\right\} = c_{bp}\sum_{k=q_p+1}^{B} k^{\eta_{bp}} \quad \text{Equation 4-19}$$

where B is the maximum queue size seen by an arrival during the measurement period. It is noted that the $queue_{high}$ region is restricted up to B packets owing to the power-law decay of the queue tail distribution model in the presence of Power-law traffic. The series in Equation 4-18 is not summable to infinity as in Equation 4-14 [7].

Substitute Equation 4-19 into Equation 4-18, we can then arrive as follows:

$$\bar{q}_{high} = \frac{\sum_{j=q_p+1}^{B} j^{\eta_{bp}+1}}{\sum_{k=q_p+1}^{B} k^{\eta_{bp}}} \quad \text{Equation 4-20}$$

In Equation 4-20, it can be noted that $\bar{q}_{high}$ constitutes two series sums. These series sums in the denominator and nominator in Equation 4-20 are comparable to the Hurwitz zeta function [GRA00], and any simpler form is analytically intractable. However, this can be approximated by a simple integration function. Consider the denominator in Equation 4-20.

---

[7] In the Markovian traffic case, we do not restrict the buffer distribution to B as the queue length distribution decays exponentially which is much faster than that of power-law traffic case. Therefore, the error introduced by not limiting the buffer distribution to B is negligible when performing convolution.

$$\sum_{k=q_p+1}^{B} k^{\eta_{bp}} \approx \int_{q_p+1}^{B} k^{\eta_{bp}}\, dk$$

$$= \frac{B^{\eta_{bp}+1} - \left(q_p+1\right)^{\eta_{bp}+1}}{\eta_{bp}+1}$$

Equation 4-21

The approximation error in Equation 4-21 depends on the shape of the curve as shown in Figure 4-13.

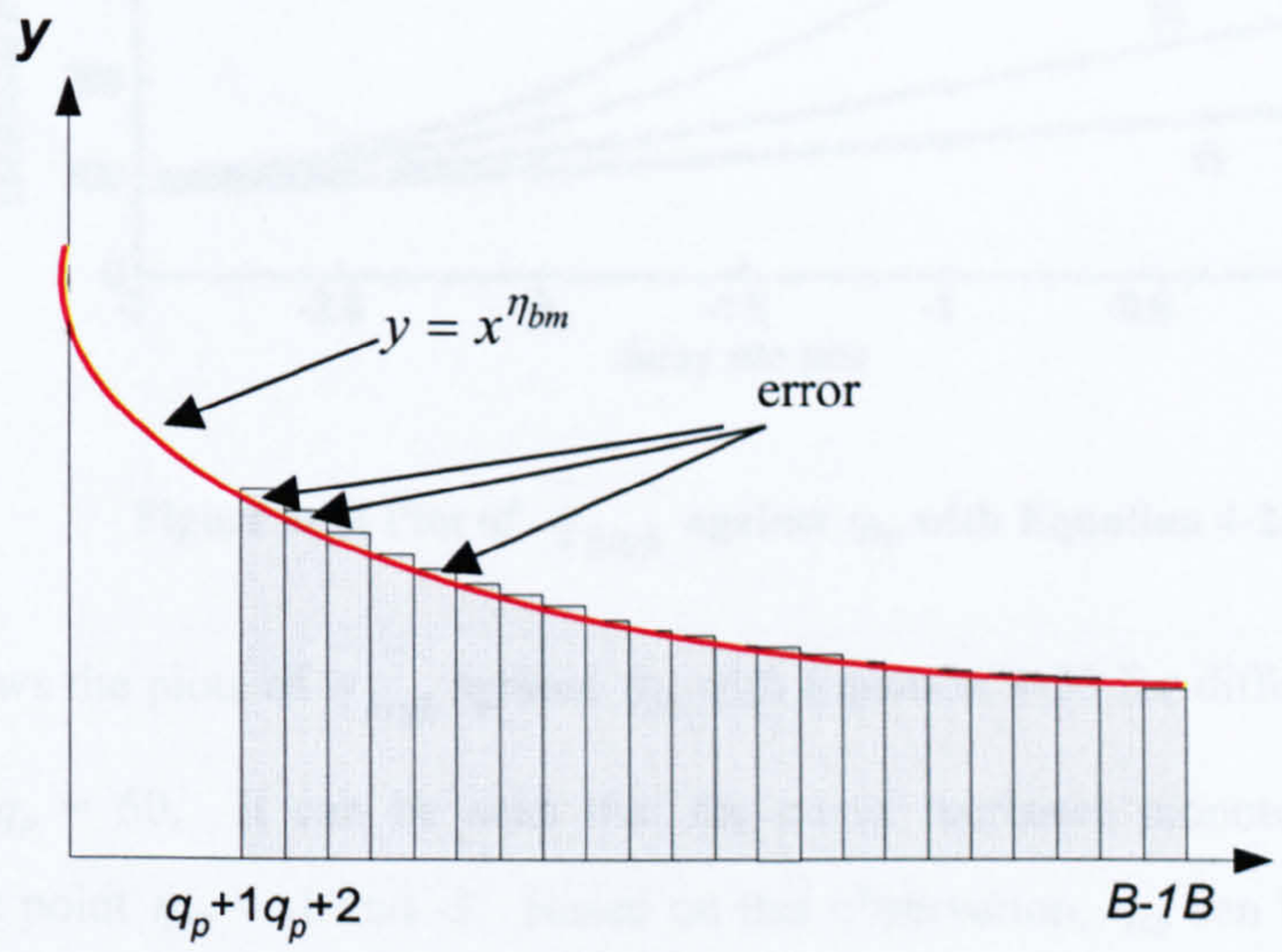


Figure 4-13 Integral approximation for series sum

In Figure 4-13, the series sum $\sum_{k=q_p+1}^{B} k^{\eta_{bp}}$ is equal to grey area. The integral $\int_{q_p+1}^{B} k^{\eta_{bp}}\, dk$ is equal to the area underneath the red curve within the range from the point q_p+1 to B. Therefore, the approximation error is the difference between these two areas. Due to the power-law decay characteristic in the tail of the queue length distribution, this approximation error is virtually negligible. Then Equation 4-19 and Equation 4-20 are approximated as follows:

$$p_{high} \approx c_{bp} \int_{q_p+1}^{B} k^{\eta_{bp}}\, dk = \frac{c_{bp}}{\eta_{bp}+1}\left(B^{\eta_{bp}+1} - \left(q_p+1\right)^{\eta_{bp}+1}\right)$$

Equation 4-22

$$\bar{q}_{high} \approx \frac{\eta_{bp}+1}{\eta_{bp}+2} \frac{B^{\eta_{bp}+2} - \left(q_p+1\right)^{\eta_{bp}+2}}{B^{\eta_{bp}+1} - \left(q_p+1\right)^{\eta_{bp}+1}}$$

Equation 4-23

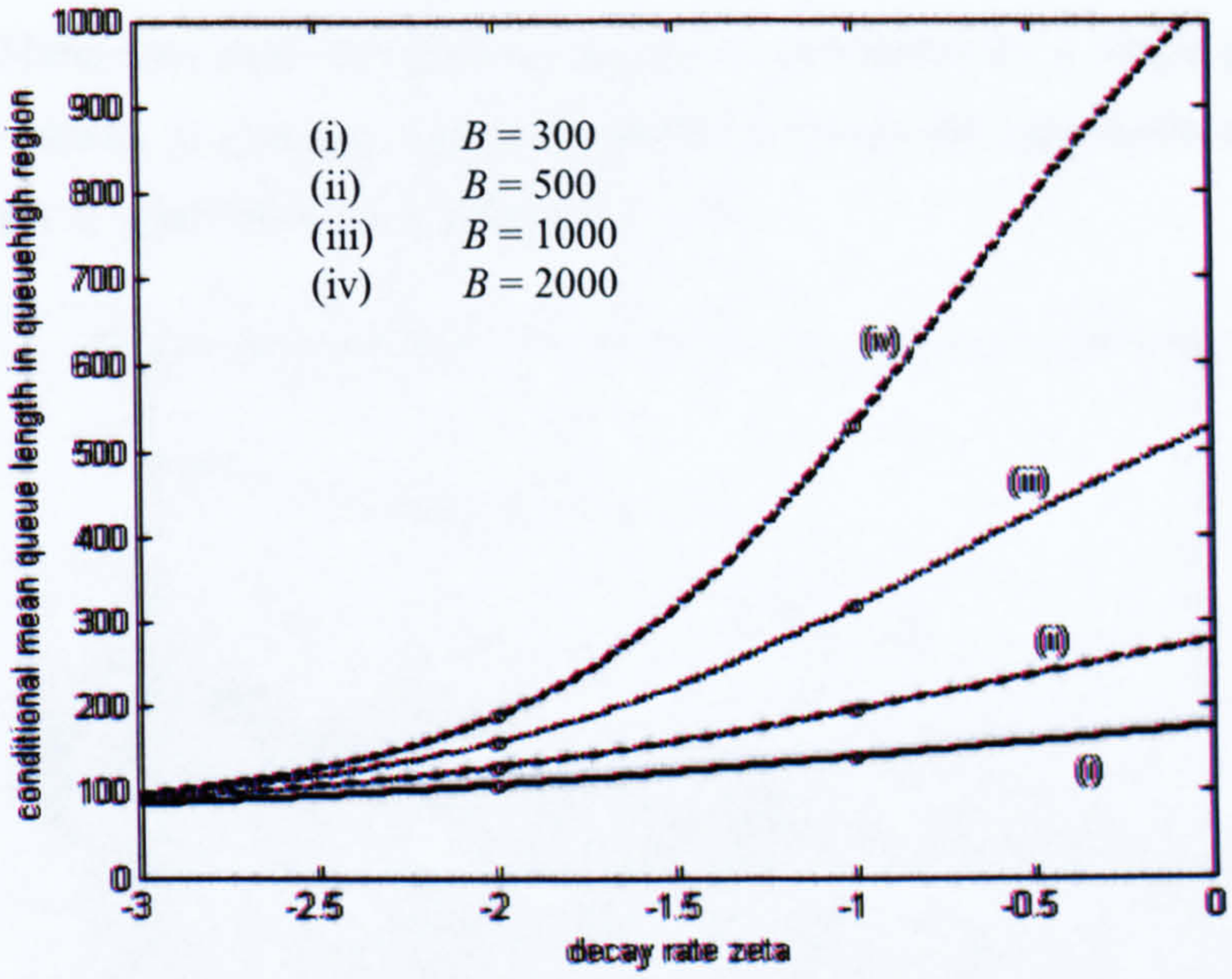


Figure 4-14 Plot of $\bar{q}_{high}$ against η_{bp} with Equation 4-23

Figure 4-14 shows the plots of $\bar{q}_{high}$ against η_{bp} with Equation 4-23 for different value B with fixed partition point q_p = 50. It can be seen that the curve increases monotonically with η_{bp}. It is discontinuous at point η_{bp} = -1 and -2. Based on this observation, η_{bp} can be determined by using numerical iteration methods, e.g. the Newton-Raphson method. Let's define the estimate of η_{bp} as $\hat{\eta}_{bp}$. Then the estimate $\hat{c}_{bp}$ can be determined by substituting the estimate $\hat{\eta}_{bp}$ into Equation 4-22 as follows:

$$\hat{c}_{bp} = \frac{\hat{\eta}_{bp} + 1}{B^{\hat{\eta}_{bp}+1} - \left(q_p + 1\right)^{\hat{\eta}_{bp}+1}} p_{high} \qquad \textbf{Equation 4-24}$$

Figure 4-15 depicts the re-construction for the queue length distribution. This can be described as follows:

- In $queue_{low}$ region, i.e. queue state $X \leq q_p$:

  $\hat{Q}_p(X = \bar{q}_{low}) = p_{low}$

  elsewhere, the queue state probability = 0
- In $queue_{high}$ region, i.e. queue state $X > q_p$:

  $\hat{Q}_p(X = x) = \hat{c}_{bp} \cdot x^{\hat{\eta}_{bp}}$

Similar to the Markovian case, the $queue_{low}$ region is represented by a single point, whereas, the queue state probability is $queue_{high}$ region is constructed with the burst-scale constant and decay estimate as shown in Equation 4-23 and Equation 4-24.

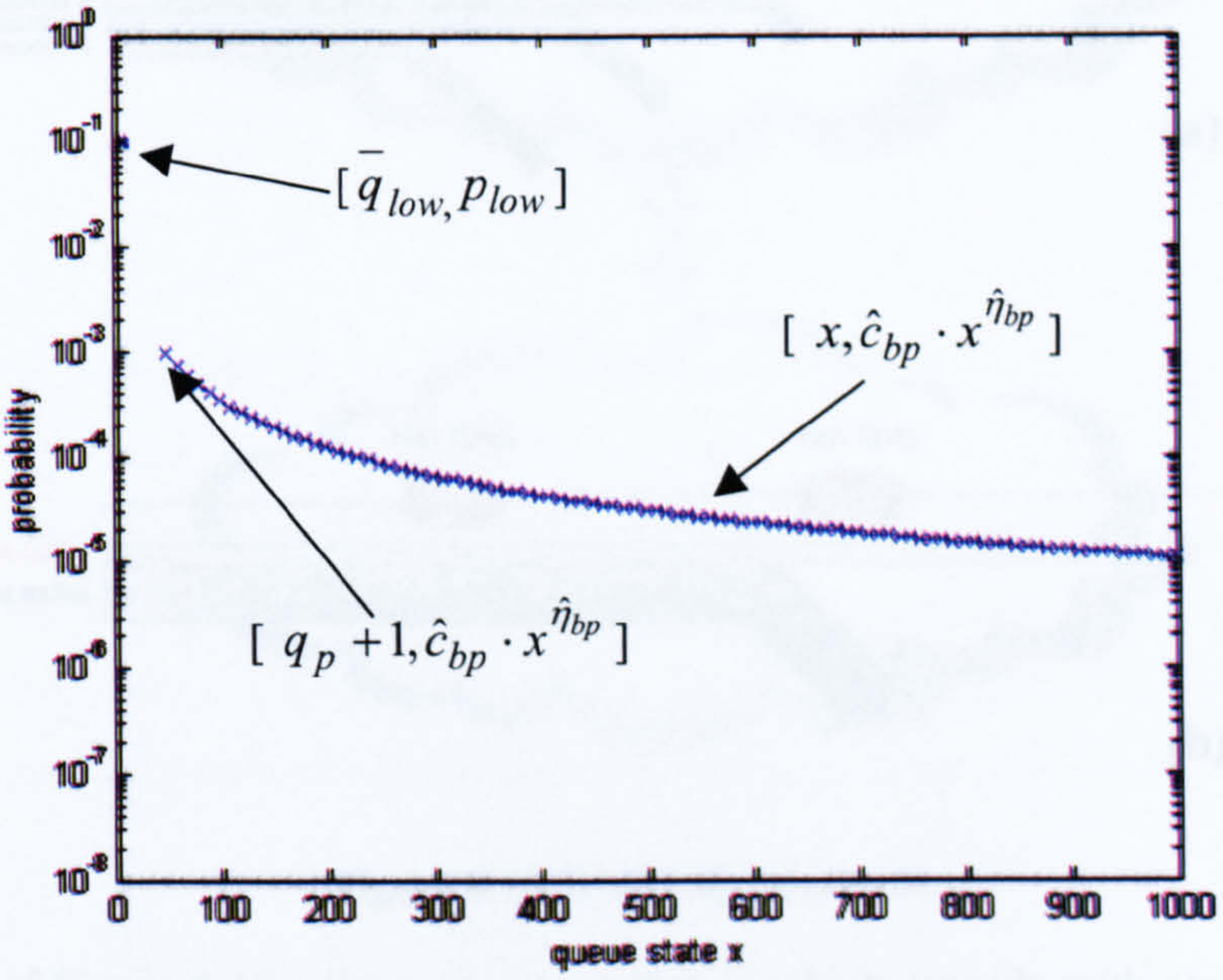


Figure 4-15 Re-construction of queue length distribution $\hat{Q}_p(x)$

4.3. End-to-end Delay Distribution Estimation – at NOC

As discussed in Section 3.3.C, the end-to-end delay distribution is estimated by using the n-fold convolution of the delay distribution at each queue as follows:

$$d_{end-to-end}(t) = d_1(t) * d_2(t) * \ldots * d_n(t) \qquad \textbf{Equation 4-25}$$

where $d_{end\text{-}to\text{-}end}(t)$ denotes the end-to-end delay distribution of a specific path, $d_k(t)$ is the delay distribution of the queue at node k. The operator * represents the continuous convolution process. The above convolution operation makes the assumption that the correlation between the delays of successive queues is negligible. Although it is impossible for the queuing delay to be completely uncorrelated for the successive queues in real practice, experimental results showed that it is fair enough to make such assumption on the queuing delays in many networks [VLE95] [SEU00]. Furthermore, the correlation of successive queuing delays is not significant e.g. in the following scenario.

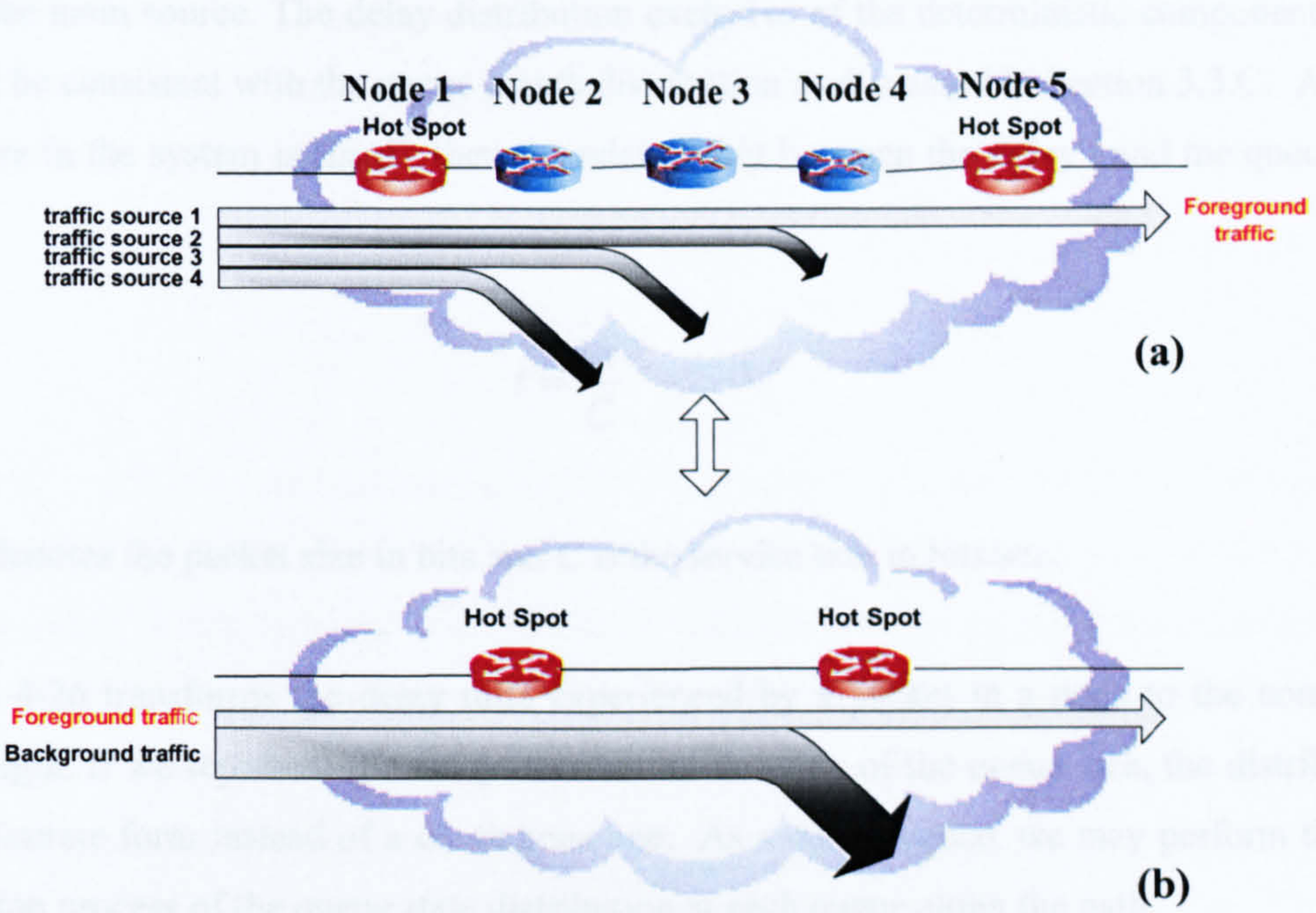


Figure 4-16 Traffic flow example

In the example of Figure 4-16a, there are two hotspots which cascade with several lightly-loaded nodes. It is also a typical scenario of traffic flow in the real Internet paths. In [RIB03], the authors pointed out that congestion largely occurs at the edge of network close to the source or the receiver, i.e. the access nodes usually are the hotspots. In [PAP02], it reports very low packet loss and queuing delay on the backbone networks. This supports the fact that congestion occurs at the edge.

In Figure 4-16a, let's assume the path taken by traffic source 1 is of interest. This traffic is often referred to as *foreground traffic* [LIU02] [STE02]. The traffic 2, 3 and 4 are competing for the resource with traffic source 1, these traffics are multiplexed into different ports in the next few hops after leaving the same queue at node 1. If we neglect the queuing delay in the core network, then Figure 4-16a can be visualized to its counterpart as shown in Figure 4-16b. Except the foreground traffic, all the traffics are routed elsewhere before entering the next hotspot. This traffic is grouped and referred to as *background traffic*. As a result, if the load of the foreground traffic is small compared with that of the background traffic, then the link-delay independence assumption will be very reasonable and provides the corresponding end-to-end distribution quite accurate with the convolution process [ØST02].

As discussed in Section 3.2, the packet delay consists of stochastic and deterministic components. The deterministic component is intrinsic and fixed for a specific path. If we want to know the stochastic component, then, measurement is a necessity. To avoid ambiguity, the delay distribution

is referred to the stochastic part for the rest of this thesis. For the stochastic component, the queuing delay is the main source. The delay distribution exclusive of the deterministic component in a local node will be consistent with the queue length distribution as discussed in Section 3.3.C. Assume the packet size in the system is fixed[8], then the relationship between the delay t and the queue length x will be:

$$t = \frac{ex}{C} \qquad \textbf{Equation 4-26}$$

where e denotes the packet size in bits and C is the service rate in bits/sec.

Equation 4-26 transforms the delay time experienced by a packet in a node to the corresponding queue length. If we represent the delay distribution in terms of the queue size, the distribution will be of a discrete form instead of a continuous one. As a consequence, we may perform the discrete convolution process of the queue state distribution at each queue along the path.

$$Q_{end-to-end}(x) = Q_1(x) \otimes Q_2(x) \otimes \ldots Q_n(x) \qquad \textbf{Equation 4-27}$$

where $Q_{end\text{-}to\text{-}end}(.)$ is the end-to-end delay distribution with respect to queue size, the operator $\otimes$ represents the discrete convolution process.

4.4 Normalization of Queue State Distribution – at NOC

4.4.1 Normalization Process for Markovian Traffic

Equation 4-27 is used to estimate the end-to-end delay distribution when the service rates for the queues along the path are the same. However, in practice, they may vary from one to another. For example, in the Class-Based Queuing scheduling or link sharing scheme, the link capacity can be partitioned and assigned a portion of the link capacity to a specific class. In view of this, we must take this into account and make the modification in Equation 4-27. Let's consider that there is a queue with the service rate C_k, here, we assume the link capacity can be precisely partitioned to different class of traffic, although the bandwidth stealing effect is possible in any link-sharing

[8] The parameter, the fixed packet size e, can be substituted by mean packet length e_v in the regime of variable packet length. This is illustrated in Section 5.3.4.

scheme. This will be discussed in Chapter 6. Since the queue length x_k at queue k will cause the corresponding delay $d_k = (e_k x_k)/C_k$ according to Equation 4-26, if x_k is in the burst-scale region, the queue state probability is given as $c_{bmk} \cdot \eta_{bmk}^{x_k}$ for the Markovian traffic as shown in Equation 4-6. This can be further expressed in terms of delay d_k as follows:

$$c_{bmk} \cdot \eta_{bmk}^{x_k} = c_{bmk} \cdot \eta_{bmk}^{\frac{C_k}{e_k} d_k} \qquad \textbf{Equation 4-28}$$

Let's define a reference service rate C_r, assume the service rate in queue k is equal to this reference service rate instead of its real service rate C_k. In order to cause the same delay d_k with service rate C_r, the queue length should be equal to x_k', where $x_k' = (d_k C_r)/e_r$. For example, if the reference service is double the true service rate, then a double size of the queue length will cause the same queuing time. Rearranging $x_k' = (d_k C_r)/e_r$, this becomes $d_k = (e_r x_k')/C_r$. We substitute this into Equation 4-28, then we can obtain the following expression by assuming the packet size in the system is fixed.

$$c_{bmk} \eta_{bmk}^{x_k} = c_{bmk} \eta_{bmk}^{\frac{C_k}{e_k} d_k} = c_{bmk} \eta_{bmk}^{\frac{e_r C_k}{e_k C_r} x_k'} = c_{bmk} \left(\eta_{bmk}^{\frac{C_k}{C_r}} \right)^{x_k'} \qquad \textbf{Equation 4-29}$$

Equation 4-29 shows that if we assume the service rate in queue k is C_r instead of the actual service rate C_k, the burst-scale decay rate will change to η_{bmk}' which is equal to $\eta_{bmk}^{\frac{C_k}{C_r}}$ from its original value η_{bmk}.

Apart from the burst-scale decay rate, it is necessary to modify the decay constant as well. In the original queue state distribution, with reference to Figure 4-12, the point q_p+1 indicates the beginning of $queue_{high}$ region. To have the same delay effect when the reference service rate is C_r instead, the queue length should be equal to $(C_r/C_k)(q_p+1)$. Let's define c_{bmk}' to be the modified burst-scale decay constant, then if we sum up the probability in the $queue_{high}$ region, it should be equal to p_{highk}.

$$c_{bmk}' \left\{ \left(\eta_{bmk}^{\frac{C_k}{C_r}} \right)^{\frac{C_r}{C_k}(q_p+1)} + \left(\eta_{bmk}^{\frac{C_k}{C_r}} \right)^{\frac{C_r}{C_k}(q_p+1)+1} + \cdots \right\} = p_{highk} \qquad \textbf{Equation 4-30}$$

After some series manipulations, we obtain the equation for modified burst-scale decay constant.

$$c_{bmk}{}' = \frac{p_{highk}\left(1-\eta_{bmk}\frac{C_k}{C_r}\right)}{\eta_{bmk}{}^{(q_p+1)}} \qquad \textbf{Equation 4-31}$$

Therefore, the modified $queue_{high}$ region will start at $(C_r/C_k)(q_p+1)$ with the modified decay constant c_{bmk}' and modified decay rate η_{bmk}'. We need to shift the position of the point that represents the $queue_{low}$ region in Figure 4-12. Similarly, we need to modify it in order to have the same delay effect when considering the queue being served with the reference service rate. Consequently, the point should be shifted from $\bar{q}_{lowk}$ to $(C_r/C_k)\bar{q}_{lowk}$ with the same probability value p_{lowk}.

This completes the normalization process for the case of Markovian traffic. The estimate of the modified decay rate $\hat{\eta}_{bmk}{}'$ in Equation 4-29 can be obtained by substituting the original decay rate estimate in Equation 4-15. Likewise, the estimate of the modified decay constant $\hat{c}_{bmk}{}'$ is determined in Equation 4-31 with the original decay rate estimate.

To achieve the normalization process, a reference service rate is firstly defined and this can be the largest or smallest service rate along the path. Then, with respect to this reference rate, the queue length distributions are modified as follows:

Queue length reconstruction at queue k, $\hat{Q}_{mk}$ (without normalization process)	Queue length reconstruction at queue k, $\hat{Q}_{mk}{}'$ (with normalization process)
a. in $queue_{low}$ region, i.e. queue state $X \le q_p$ $\hat{Q}_{mk}(X=\bar{q}_{lowk}) = p_{lowk}$ elsewhere, queue state probability = 0	a. in new $queue_{low}$ region, i.e. queue state $X \le (C_r/C_k)q_p$ $\hat{Q}_{mk}{}'(X=(C_r/C_k)\bar{q}_{lowk}) = p_{lowk}$ elsewhere, queue state probability = 0
b. in $queue_{high}$ region, i.e. queue state $X > q_p$ $\hat{Q}_{mk}(X=x) = \hat{c}_{bmk}\cdot\hat{\eta}_{bmk}{}^{x}$	b. in new $queue_{high}$ region, i.e. queue state $X > (C_r/C_k)(q_p+1)$ $\hat{Q}_{mk}{}'(X=x) = \hat{c}_{bmk}{}'\cdot\hat{\eta}_{bmk}{}'^{x}$

Table 4-1 Summary of the queue state reconstruction with or without normalization

The end-to-end delay distribution can be obtained by convolving the normalized queue state distribution $\hat{Q}_{mk}(x)$ as in Equation 4-32.

$$Q_{end-to-end}(x) = \hat{Q}_{m1}{}'(x) \otimes \hat{Q}_{m2}{}'(x) \otimes ... \hat{Q}_{mn}{}'(x) \qquad \textbf{Equation 4-32}$$

4.4.2 Normalization Process for Power-law Traffic

In this section, the normalization process for Power-law traffic is discussed. Similar to the approach in Markovian traffic case, the queue length distribution is modified in such a way that it reflects the same delay distribution with respect to a reference service rate. The derivation is the same as illustrated in the previous section. A queue k is served with service rate C_k, queue size x_k will give delay time $d_k = (e_k/C_k)x_k$. x_k' is the queue size which cause the same amount of queuing delay if we assume service rate is C_r instead. Based on the heavy-tailed model, we have the following expression.

$$c_{bpk} \cdot x_k{}^{\eta_{bpk}} = c_{bpk}\left(\frac{C_k}{e_k} d_k\right)^{\eta_{bpk}} = c_{bpk}\left(\frac{e_r C_k}{e_k C_r} x_k{}'\right)^{\eta_{bpk}} = c_{bk}{}' \cdot x_k{}'^{\eta_{bpk}} \qquad \textbf{Equation 4-33}$$

With reference to Equation 4-33, this shows that the decay rate remains unchanged after the normalization process. However, the q_{lowk}, q_{pk} and B_k must be shifted to $(C_r/C_k)q_{lowk}$, $(C_r/C_k)q_{pk}$ and $(C_r/C_k)B_k$ respectively to have the same delay effect. Since the new $queue_{highk}$ region starts at $(C_r/C_k)(q_p+1)$ and this region has the probability of occurrence equal to p_{highk}, the decay constant should therefore be adjusted as follows:

$$p_{highk} = c_{bpk}{}' \left\{ \left[\left(\frac{C_r}{C_k}\right)(q_p+1) \right]^{\eta_{bpk}} + \left[\left(\frac{C_r}{C_k}\right)(q_{pk}+1)+1 \right]^{\eta_{bpk}} + ... \left[\left(\frac{C_r}{C_k}\right) B_k \right]^{\eta_{bpk}} \right\}$$

$$= c_{bpk}{}' \sum_{j=(C_r/C_k)(q_p+1)}^{(C_r/C_k)B_k} j^{\eta_{bpk}} \qquad \textbf{Equation 4-34}$$

Using the integral approximation for the series sum as in Equation 4-21, then Equation 4-34 can be approximated as:

$$c_{bpk}{}' = \frac{1}{\dfrac{(C_r/C_k)B_k}{\sum_{j=(C_r/C_k)(q_p+1)} j^{\eta_{bpk}}}} p_{highk} \approx \frac{\eta_{bpk}+1}{\left(\dfrac{C_r}{C_k}\right)B_k{}^{\eta_{bpk}+1} - \left(\dfrac{C_r}{C_k}\right)(q_p+1)^{\eta_{bpk}+1}} p_{highk} \qquad \textbf{Equation 4-35}$$

By comparing with Equation 4-24, we arrive at:

$$c_{bpk}{}' = \frac{C_k}{C_r} c_{bpk} \qquad \textbf{Equation 4-36}$$

Based on the estimate of the decay rate and decay constant, the queue length reconstruction and the normalization process for the Power-law traffic is summarised as follows:

Queue length reconstruction at queue k, $\hat{Q}_{pk}$ (without normalization process)	Queue length reconstruction at queue k, $\hat{Q}_{mk}{}'$ (with normalization process)
a. in $queue_{low}$ region, i.e. queue state $X \le q_p$ $\hat{Q}_{pk}(X = \bar{q}_{lowk}) = p_{lowk}$ elsewhere, queue state probability = 0	a. in new $queue_{low}$ region, i.e. queue state $X \le (C_r/C_k)q_p$ $\hat{Q}_{pk}{}'(X = (C_r/C_k)\bar{q}_{lowk}) = p_{lowk}$ elsewhere, queue state probability = 0
b. in $queue_{high}$ region, i.e. queue state $X > q_p$ $\hat{Q}_{pk}(X = x) = \hat{c}_{bpk} \cdot x^{\hat{\eta}_{bpk}}$	b. in new $queue_{high}$ region, i.e. queue state $X > (C_r/C_k)(q_p+1)$ $\hat{Q}_{pk}{}'(X = x) = \hat{c}_{bpk}{}' \cdot x^{\hat{\eta}_{bpk}}$

Table 4-2 Summary of the queue state reconstruction with or without normalization

The end-to-end delay distribution can be obtained by convolving the normalized queue state distribution $\hat{Q}_{pk}(x)$.

$$Q_{end-to-end}(x) = \hat{Q}_{p1}{}'(x) \otimes \hat{Q}_{p2}{}'(x) \otimes \ldots \hat{Q}_{pn}{}'(x) \qquad \textbf{Equation 4-37}$$

4.5 Summary

In this Chapter, we presented our QL Measurement scheme. The goal of this measurement scheme is to capture the per-hop delay distribution. In our measurement framework, the measurement task QL Measurement is distributed to every node. A network management centre exists and the role of this element is to collect the measurement data for analysis. This involves (1) re-construction of per-hop queue length distribution. (2) Queue length distribution normalization in case of various bandwidth allocated along the path. The per-hop queue length distribution can serve for the identifying the QoS degradation in the network. Additionally, the per-hop distributions can be used to determine the end-to-end delay distribution.

QL Measurement scheme is deliberately designed in the simplest way. The measurement process only consists of the addition and the comparison process and so the routing/switching efficiency of the router/switch will not be affected. Furthermore, there is only five measurement data for each measurement. Therefore, it minimizes the capacity of data storage and the data transfer between the local node and the NOC.

To re-construct per-hop queue length distribution, our approach is known as model-based. We exploit the queue length characteristic for different traffic scenarios. For the Markovian traffic, the queue tail will exhibit exponential decay, whereas, it has Power-law decay for Power-law traffic. With the model-based approach, it makes the measurement methodologies simple and provides good estimation results.

Chapter 5 Evaluation of QL Measurement Scheme

In this chapter, our proposed QL Measurement scheme is evaluated through a series of simulation experiments. This chapter is divided into two sections. The first section is to assess the effectiveness of our scheme in capturing the local queue length distribution. The goal of the second section is to validate our scheme on the end-to-end delay distribution estimation.

5.1 Simulation Model

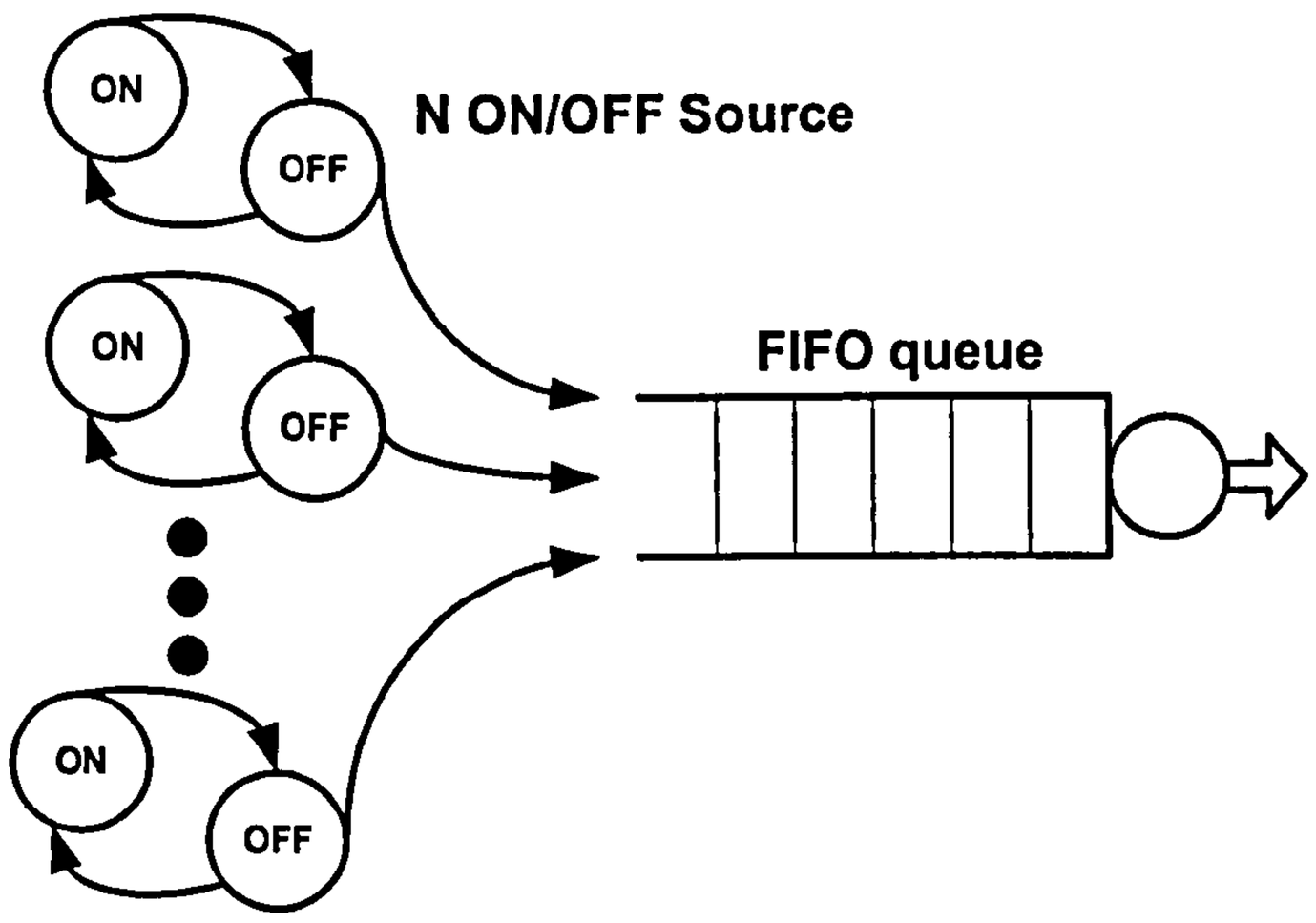


Figure 5-1 Simulation model

As explained in Chapter 4, the ON/OFF models are chosen to generate a traffic trace. Figure 5-1 depicts our simulation model. Superposition of *N* ON/OFF sources is used to produce the traffic. To generate Markovian traffic, the ON and OFF periods are governed by *exponential* distribution, whereas, the heavy-tailed *Pareto* distribution is used to model high variability ON/OFF period [TAQ97] in order to generate Power-law traffic.

5.2 An Issue of Measurement Time

The duration of measurement has long been a debatable issue for any measurement scheme. It is usually hard to predict for how long a measurement should be carried out to achieve the desired level of accuracy [JAI02]. The amount of available statistics affects the accuracy of the measurement scheme. It implies that an unreasonable long measurement time may be required for many measurement schemes [MAL97] [AID02].

The concept of a *busy hour* is important both in voice and data networks [ROU99]. Observations of traffic on network links typically reveals the intensity levels (in bits/sec) averaged over periods of several ten of minutes which are relatively predictable from day to day. Systematic intensity variations occur within the day reflecting user activity. The traffic intensity in the busy (high utilization) or non-busy period (medium or low utilization) is roughly constant [ROB01]. These periods are in the order of an hour and this constancy suggests that the traffic in Internet, telephony network, can be modelled as a stationary stochastic process [ROB01]. As a result, in many traffic situations the changing conditions or non-stationarity can be safely ignored since over the time scales of interest, they have little effect and the process may be well approximated by a stationary model [HUE98] [ROU99] [SAL01].

Based on the concept of a busy hour and the evidence supporting stationary traffic modelling either in the busy period or in non-busy period, we assume the measurement period is one hour throughout the thesis[9]. In addition, the arrival processes are stationary over the low, medium and high utilization duration [LUC97], the number of traffic sources N is kept constant during measurement to maintain the constant traffic intensity. Inspired by the busy hour concept, we perform our measurement scheme for an hour, the data collected is used to deduce the local or end-to-end delay distribution.

5.3 Per-hop Queue Length Measurement (Markovian Traffic)

To simulate the Markovian traffic, we use the exponential distribution to model the ON and OFF period of each traffic source. We use the voice traffic parameters to configure the ON and OFF durations. The typical value of ON period is 0.69 second, whereas 1.69 second for the OFF duration [PIT00].

[9] This is also recommended in [CIS03a] that measurement should be carried out over an hour.

We assume that the link bandwidth is partitioned by using a scheduling mechanism and the exact bandwidth allocation can be achieved. For brevity, we neglect the bandwidth stealing effect [10] between different classes. This will be discussed in Chapter 6.

5.3.1 Partition Point q_p Selection

In our QL Measurement scheme, the queue is logically partitioned by a partition point called q_p for measurement (see Figure 4-11). We performed simulations with *ns*-2 [BRE00] to study the impact of the partition point on the queue state distribution estimation. The parameters of ON/OFF sources are as follows: the transmission rate during ON time is 167 packet/sec, with fixed packet size = 53 byte. These are typical parameters for digital voice [STE02]. The experimental set-up is shown in the following table.

ON time	0.96sec (exponential)
OFF time	1.69sec (exponential)
Number of sources	100
bit rate	70808bps
service rate	2.85Mbps

Table 5-1 Experimental set-up

We perform our QL Measurement with different partition point value for the measurement period = 3600 seconds. The utilization factor in this experiment is 0.9 to ensure we experience burst-scale queuing behaviour. Figure 5-2 shows the simulation results of the queue length distributions. The actual queue length distribution seen by arrivals is represented by the solid red line. This illustrates that the burst-scale and packet–scale queuing region do exist in the distribution. Both of them tend to follow a straight line in a log-linear plot, i.e. exponentially decay occurs in both regions.

[10] When the bandwidth is allocated by a scheduler, the spare unused bandwidth will be shared by the non-empty queue. This effect makes the actual received bandwidth differ from the the theoretical allocated bandwidth

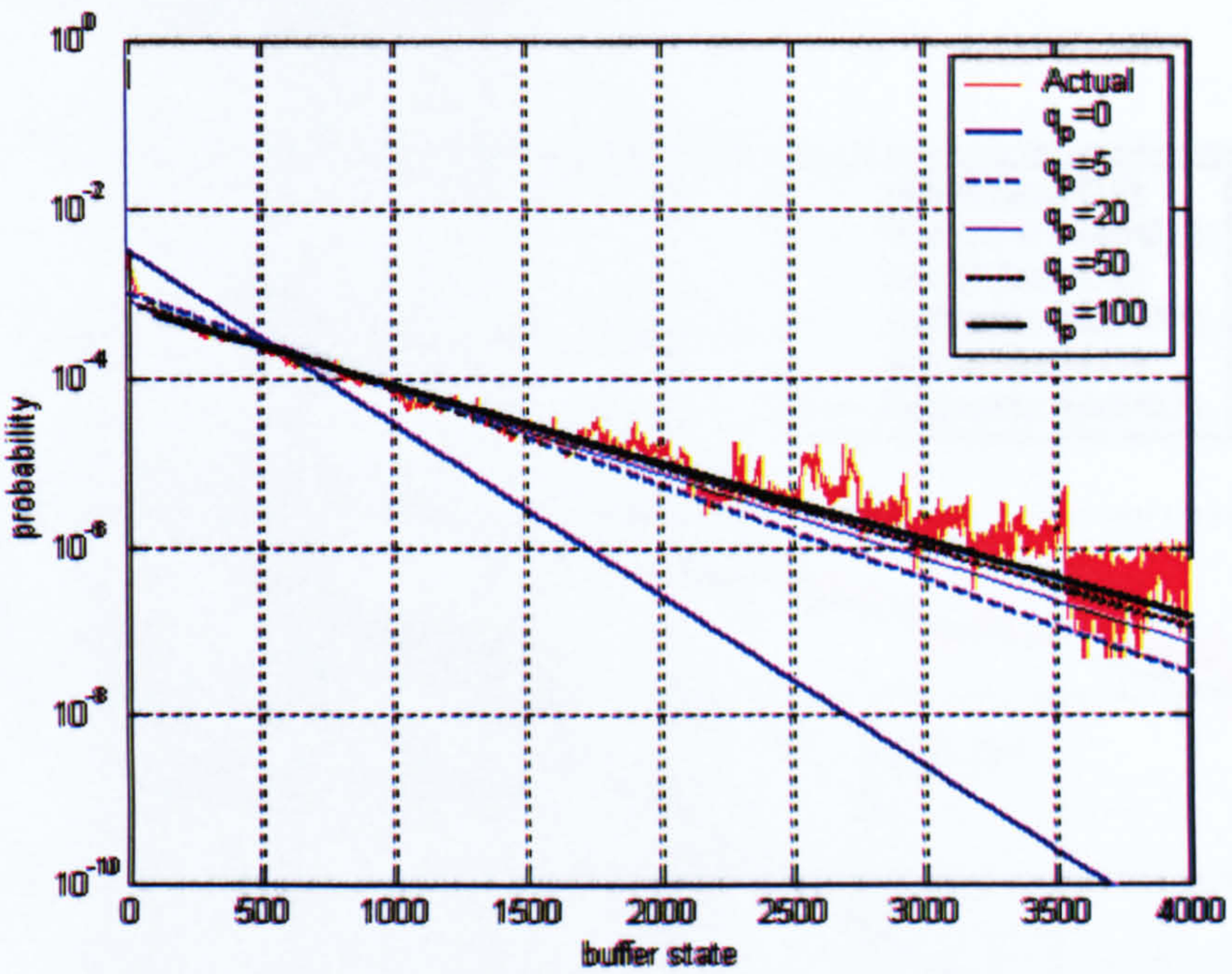


Figure 5-2 Actual queue length distribution and estimated with different partition point

Comparing the estimated queue length distributions with the actual one, it can be found that the larger the partition point is, the better estimation result is obtained. When the partition point q_p is small (0, 5), it is still in the packet-scale region and so this affects the estimation result. Once the partition point is large enough, when it is greater than 20, then the estimation results are close to the actual distribution. This suggests that the location of the partition point is not so significant provided that it is beyond the packet-scale region. Although a large value ensures that the partition point is placed in the burst-scale region, the partition point should not be unreasonably large, as it will affect the number of occurrence in the $queue_{high}$ region and so the measurement data becomes less reliable. In [PIT00], the authors state that the packet-scale region is normally within the range of the order of a few 10's of packets. As a rule of thumb, the partition point should be within the range that is larger than 20 but smaller than 100. It may be hard for the network operator to select the partition point in advance. Therefore, this property eases the task of selecting the partition point. The partition point can be arbitrary chosen to be within the range recommended above. Then, the estimation result will not be sensitive to the partition point.

5.3.2 Different Load Condition

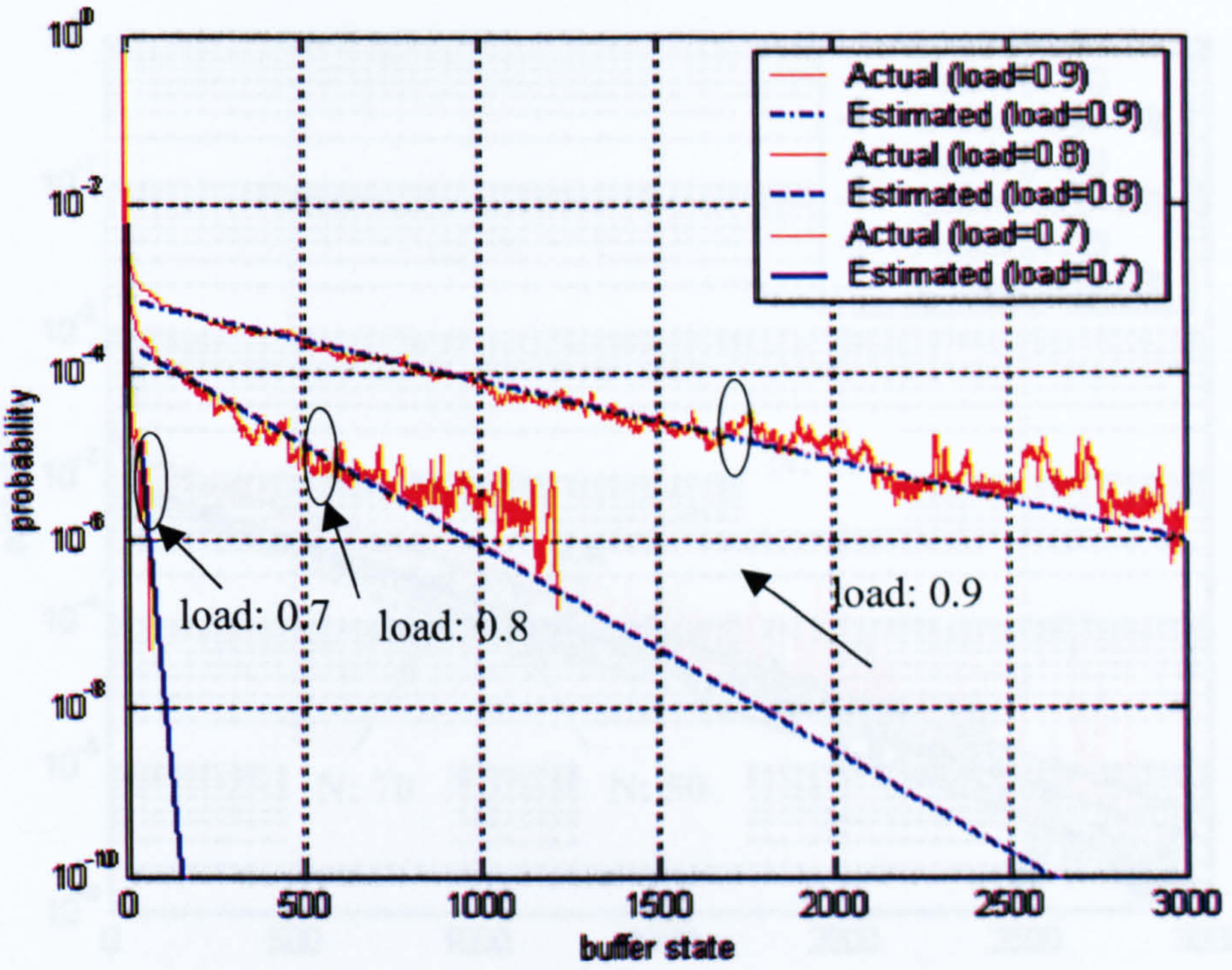


Figure 5-3 Actual queue length distribution and estimated with different load condition

Figure 5-3 shows the comparison results between the actual queue length distributions with the estimated one with our QL scheme under different load condition. The experimental set-up is the same as that in Table 5.1. The service rate is adjusted to achieve different load conditions.

With reference to Figure 5-3, this shows that a longer queue tail happens for a higher load condition. With a higher load, the burst-scale queuing effect becomes more significant, the gradient in the burst-scale region becomes flatter. In addition, the knee point between the burst-scale region and packet-scale region moves upwards.

As discussed in the previous section, the partition point should be within the range from 20 to 100. In this experiment, we perform QL Measurement scheme with a partition point = 50. It can be found that QL Measurement scheme is capable of capturing the queue length distribution. The estimated queue length distributions overlap the actual distributions.

5.3.3 Different Number of Multiplexed Sources

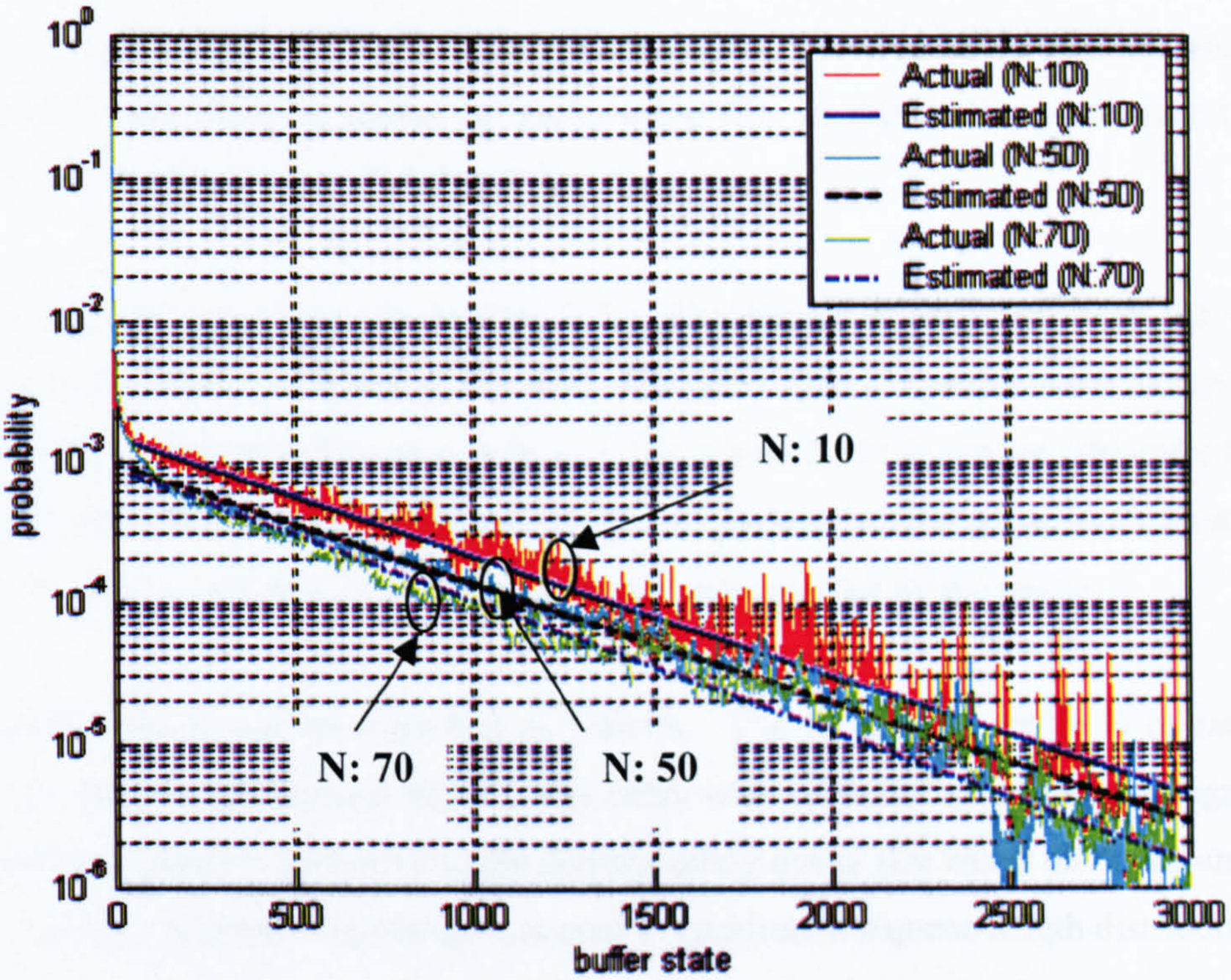


Figure 5-4 Actual and estimated queue length distribution with different number of sources

In this section, we investigate the effect of different numbers of ON/OFF sources on the queuing behaviour. The traffic is generated by superposition of different number of ON/OFF sources N=10, N=50 and N=70 respectively. The service rate is adjusted to achieve 0.9 load.

With reference to Figure 5-4, this shows that the knee point moves downwards and the tail becomes slightly steeper when the number of ON/OFF sources increases. This suggests that a shorter queue length is expected when it is multiplexing a larger number of sources under the same buffer load condition. This phenomenon is known as statistic multiplexing gain [STE02].

The estimated queue length distributions are obtained by QL Measurement with partition point = 50. Again, it can be noted that the estimated and the actual queue length distributions are closely matched.

5.3.4 Variable Packet Size

In the previous experiments, fixed packet size is assumed. Other than ATM systems, variable packet size is feasible and found in packet networks using IP. In this section, we study the effect of variable packet sizes on our methodology.

In Section 4.3, we argue that the per-hop delay distribution is consistent with the queue delay distribution in the regime of fixed packet size. The delay time t is proportional to the queue size x seen by arrivals as shown in Equation 4-26 as $x \cdot C/e$, where C is the service rate and e is the packet size in the system. However, in the regime of variable packet size, the queue size x on average gives the delay time $x \cdot C/e_v$, where e_v is the average packet size received by the queue.

The simulation set-up can be described as follows. The foreground traffic is of interest and is multiplexed with the background traffic. The delay time experienced by every foreground packet after traversing a queue is converted to the corresponding queue size based on the mean packet size. Then this statistic, in terms of queue size, is used to construct the queue length distribution.

We performed the simulation for the following scenarios:

- Packet Size with uniform distribution
- Packet Size with trimodal distribution [AGI00].

5.3.4.1 Packet Size with Uniform Distribution

In this experiment, the queue is multiplexed by one foreground traffic and 69 background traffics. The ON/OFF sources' parameters are as shown in Table 5.1. The load is 0.85. [AGI00] reports that the minimum and maximum packet size in IP traffic is 40 byte and 1500 byte respectively. We assume the packet size is not fixed but uniformly distributed from 40 byte to 1500 byte instead. Therefore, the mean packet size is 770 bytes.

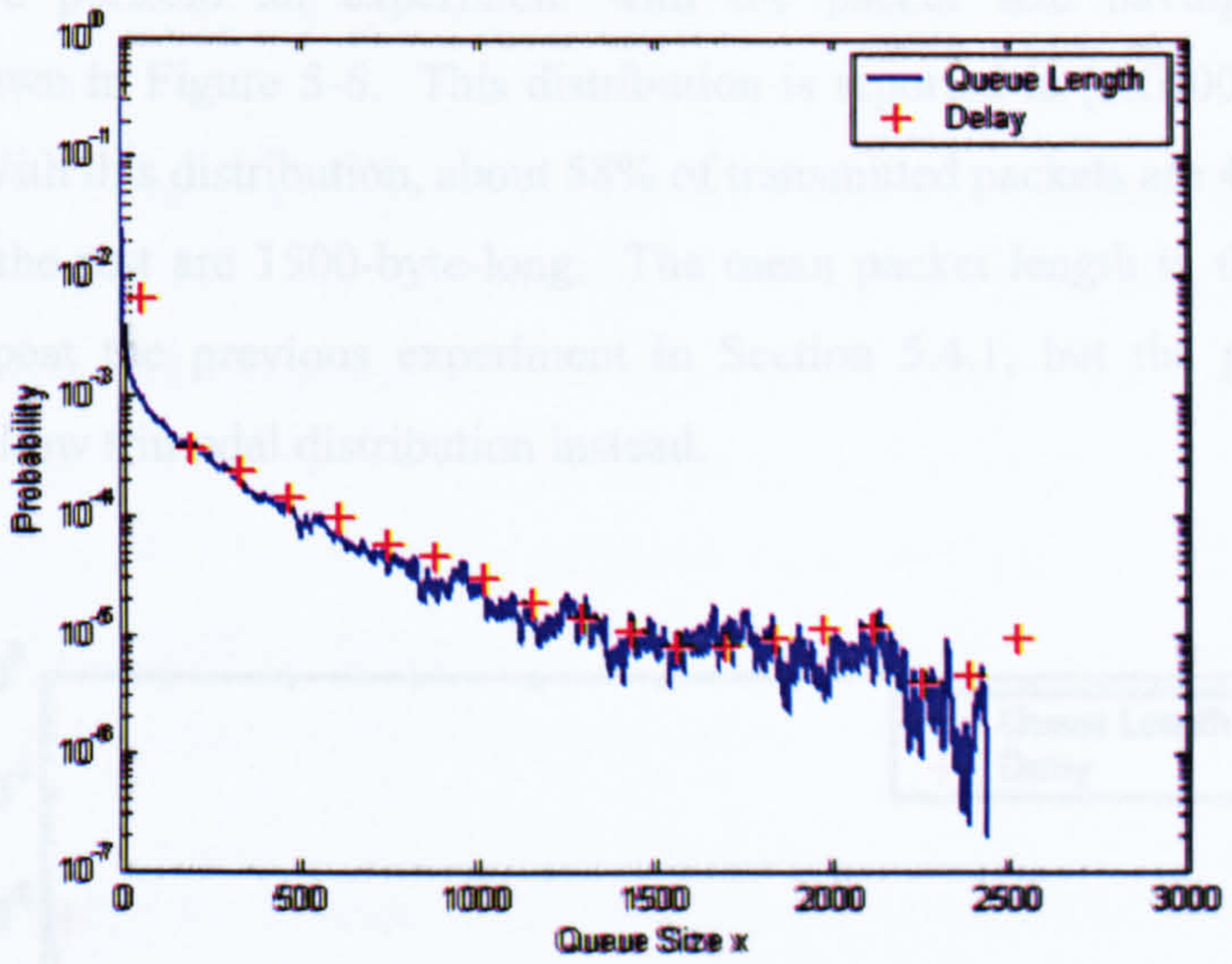


Figure 5-5 Comparison between delay and queue length distribution for variable packet size

For the purpose of comparison, we collected and measured all the foreground traffics' delay time. This statistic is converted to the queue length distribution [11] with respect to the mean packet size of 770 bytes. Figure 5-5 shows the comparison between the actual queue length distribution and the delay distribution with respect to the mean packet size. The delay distribution is binned and the average value in the bin is plotted in order to reduce the variance of the graph to have better visualisation. This technique is also employed in the work [MA03]. The result shows that two distributions agree with each other. This suggests that the mean packet size is representative in a regime of variable packet size. The mean packet size is useful to relate the delay time and the waiting queue size.

5.3.4.2 Packet Size with Trimodal Distribution

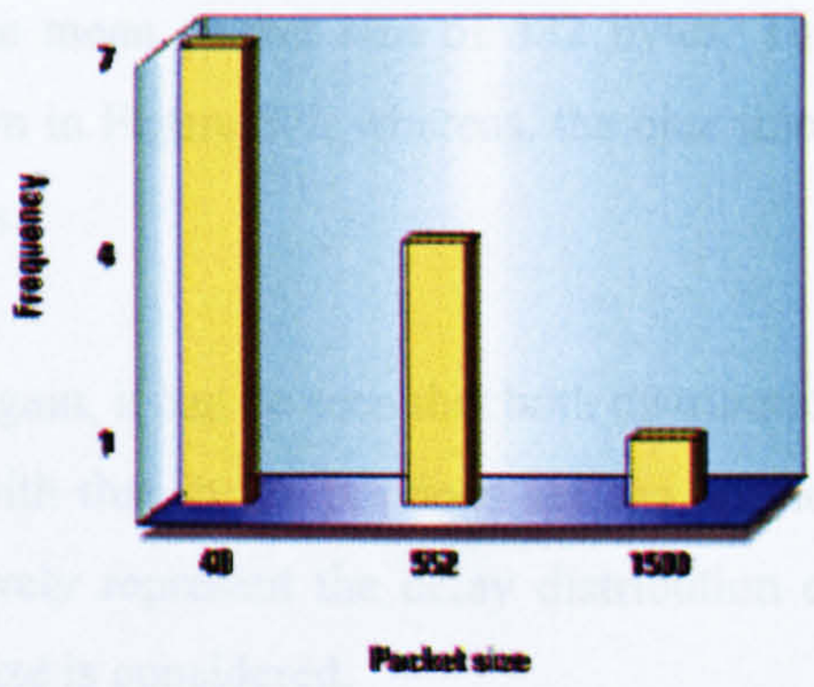


Figure 5-6 7,4,1 distribution [AGI00]

[11] It is noted that the delay distribution is referred to probability density function (pdf) as it is continuous in nature. However, after mapping delay distribution to queue length distribution by using Equation 4-26, then the distribution will be discrete and so it is referred as probability mass function (pmf).

In this section, we perform an experiment with the packet size having a trimodal (7,4,1) distribution[12] as shown in Figure 5-6. This distribution is reported in [AGI00] and occurs in most packet networks. With this distribution, about 58% of transmitted packets are 40-byte-long, 33% are 552-byte-long and the rest are 1500-byte-long. The mean packet length in this case will be 332-byte-long. We repeat the previous experiment in Section 5.4.1, but the packets generated by ON/OFF sources follow trimodal distribution instead.

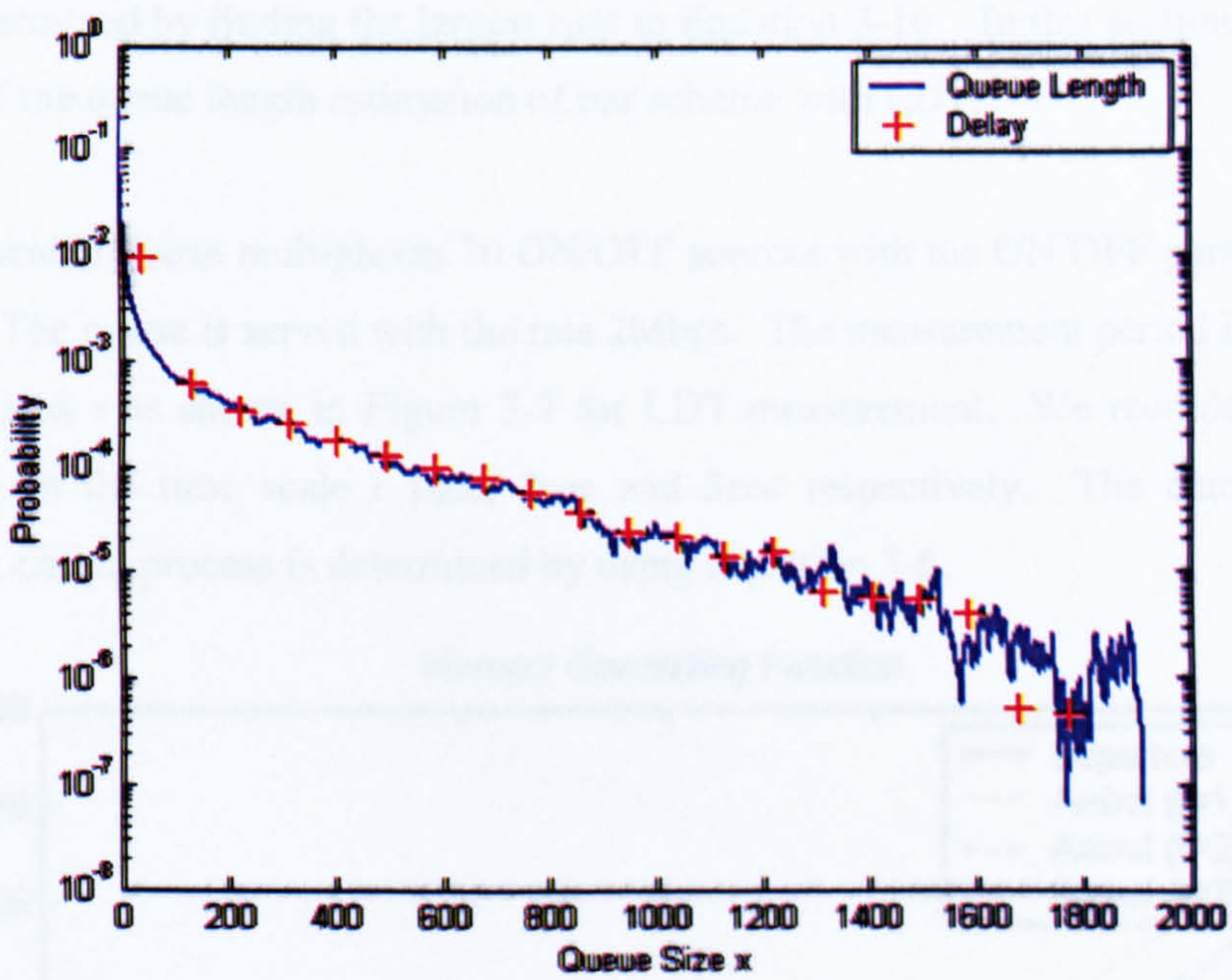


Figure 5-7 Comparison between delay and queue length distribution for trimodal distribution

Similar to Section 5.3.4.1, we converted the foreground traffics' delay statistic to the queue length distribution with respect to the mean packet size of 332 bytes. This is represented by the curve marked by red crosses as shown in Figure 5-7, whereas, the blue line shows the actual queue length distribution seen by the arrivals.

With reference to Figure 5-7, again, it can be seen that both distributions agree with each other. This experimental result together with that in the previous section provide the evidence that the queue length distribution can effectively represent the delay distribution even in the regime of variable packet length if mean packet size is considered.

---

[12] Please refer to Appendix D for the simulations for variable packet size with bimodal distribution.

5.3.5 Comparison with Large Deviation Theory

The principle idea of Large Deviations reveal that the queue tail distribution exhibits exponential decay for a wide class of traffic. The Large Deviation Theory (LDT) has been extensively studied in the application of measurement-based CAC [DUF98] [CHAN00]. The queue tail distribution is estimated by measuring the traffic arrivals as discussed in Section 3.4.1.2.A. By measuring the cumulant generating function for the traffic arrival as shown in Equation 3-6, the burst-scale decay rate can be determined by finding the largest root in Equation 3-10. In this section, we compare the performance of the queue length estimation of our scheme with LDT.

In this experiment, a queue multiplexes 70 ON/OFF sources with the ON/OFF parameters as shown in Table 5.1[13]. The queue is served with the rate 2Mbps. The measurement period is subdivided into smaller time block t as shown in Figure 3-7 for LDT measurement. We recorded the volume of traffic arriving in the time scale t 1sec, 2sec and 3sec respectively. The cumulant generating function of the arrival process is determined by using Equation 3-6.

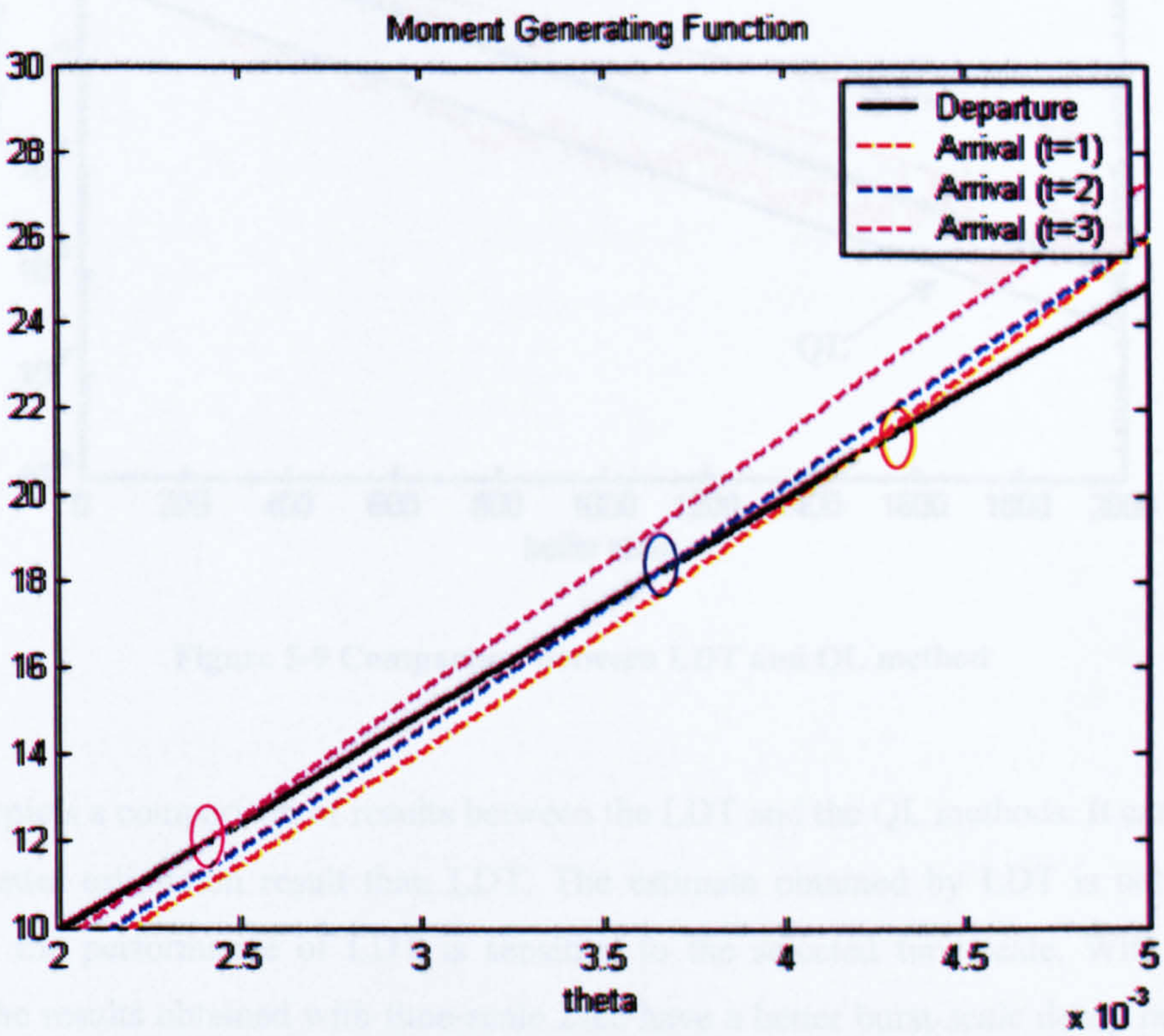


Figure 5-8 Cumulant generating functions $\Lambda_A(\theta)$, $\Lambda_B(\theta)$

[13] the exponential decay property is not changed by different amount of ON/OFF sources, see Figure 5.4

The cumulant generating function for the departure is given as Equation 3-8. Figure 5-8 shows the cumulant generating functions for the arrivals (dashed lines) and the departures (solid black line). A straight line is obtained for the departure's cumulant generating function owing to the fixed service rate. [CHAN00] gives the proof that the cumulant generating function is generally convex in shape. Therefore, two cumulant generating functions (arrival and departure) meet at two intercepting points which represent the roots in Equation 3-10. As mentioned in Section 3.4.1.2.A, the largest root θ^* (as circled in Figure 5-8) gives the solution of the decay rate as $e^{-\theta^*}$ (see Equation 3-11). According to Figure 5-8, the values of θ^* are 0.004218, 0.00362, 0.00245 for t = 1sec, 2sec and 3sec respectively. The tail of the queue length distribution is then estimated with Equation 3-12.

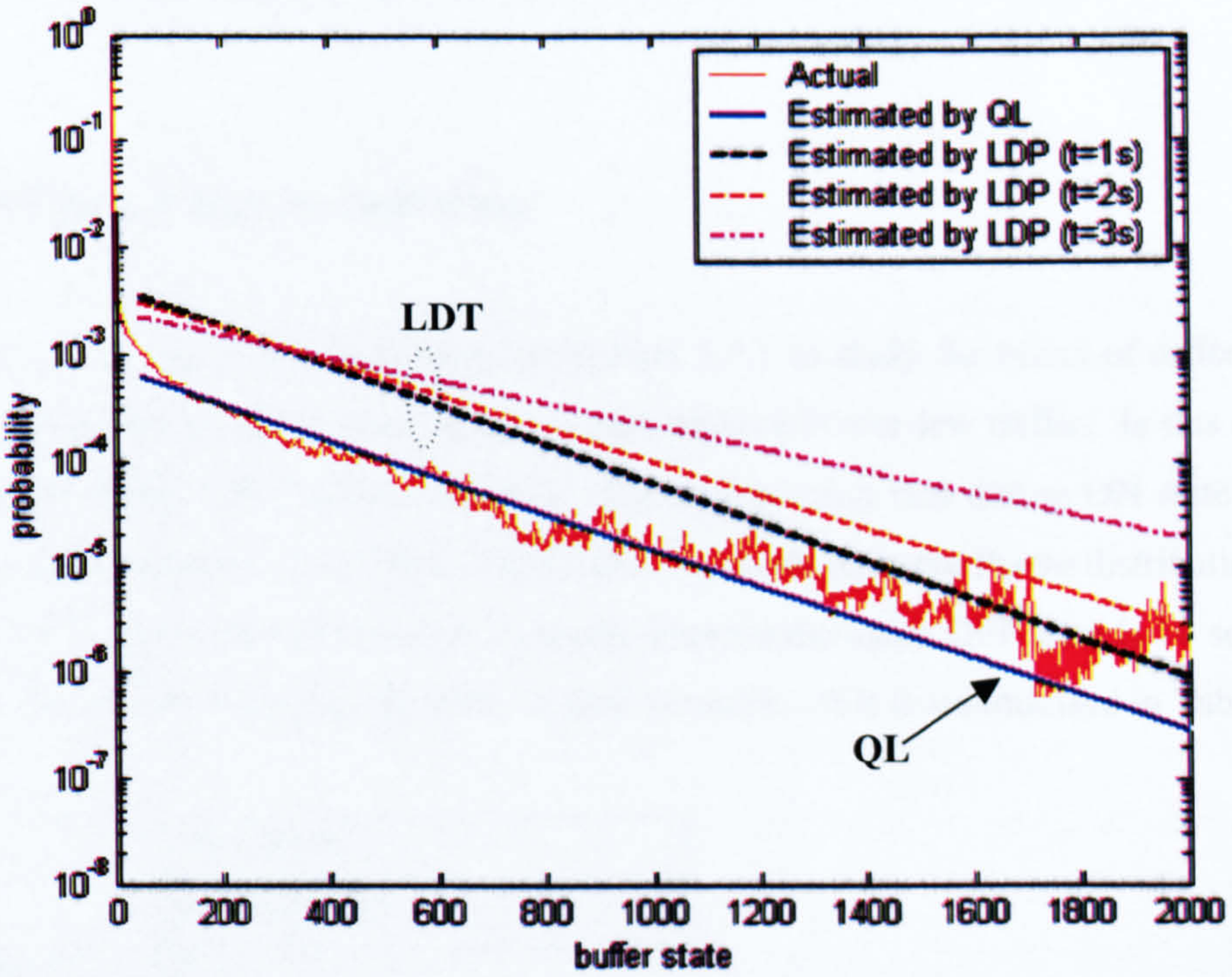


Figure 5-9 Comparison between LDT and QL method

Figure 5-9 depicts a comparison of results between the LDT and the QL methods. It can be seen that QL shows better estimation result than LDT. The estimate obtained by LDT is not satisfactory. Furthermore, the performance of LDT is sensitive to the selected time-scale. With reference to Figure 5-9, the results obtained with time-scale 2sec have a better burst-scale decay rate estimation compared with time-scale 1sec and 3sec. Since it is hard to predict what the best time-scale is, it may therefore be very time-consuming to search for the best time-scale. Otherwise, the result given by LDT is not guaranteed.

5.4 Per-hop Queue Length Measurement (Power-law Traffic)

To generate Power-law traffic, the ON/OFF traffic model is used again with the rationale discussed in Section 4.1.2. Compared with Markovian traffic used in Section 5.3, the ON/OFF periods have the heavy-tailed Pareto distribution instead. [TAQ97] reports that self-similar or long-range dependent network traffic can be generated by superposition of multiple ON/OFF sources with Pareto-distributed ON and OFF periods. The shape parameter of the Pareto distribution in Equation 4-1 should be between 1 and 2. The lower the value of the shape parameter, the higher the probability of an extremely large duration ON or OFF period as discussed in Section 4.1.1.2. In this section, we will examine QL Measurement in the presence of Power-law traffic.

5.4.1 Partition Point q_p Selection

In this section, we repeat the experiment in Section 5.3.1 to study the effect of different values of partition point for the scenario that a queue is multiplexing Power-law traffic. In this experiment, a queue is multiplexing 100 ON/OFF sources. The transmission rate during ON state is 1500kbps. The packet size is fixed as 1500 bytes. The ON/OFF periods have the Pareto distribution with shape parameter 1.4 [14]. The mean ON time is 1 second whereas the mean OFF time is 10 seconds. These parameters are chosen to mimic the traffic in data network. This is summarised in Table 5.2.

ON time	1sec (Pareto)
OFF time	10sec (Pareto)
Shape parameter	1.4
Number of sources	100
Packet Size	1500byte
bit rate	1500kbps
service rate	25Mbps

Table 5-2 Experimental set-up (Power-law)

We performed QL Measurement for Power-law traffic as explained in Section 4.2.3.2 for a measurement period 3600 second with partition point =5, 50 and 100. Based on the measurement data, we estimated the burst-scale decay rate and decay constant and used this to compare with the actual queue length distribution.

[14] Simulation result for different shape parameters is given in Section 5.4.3.

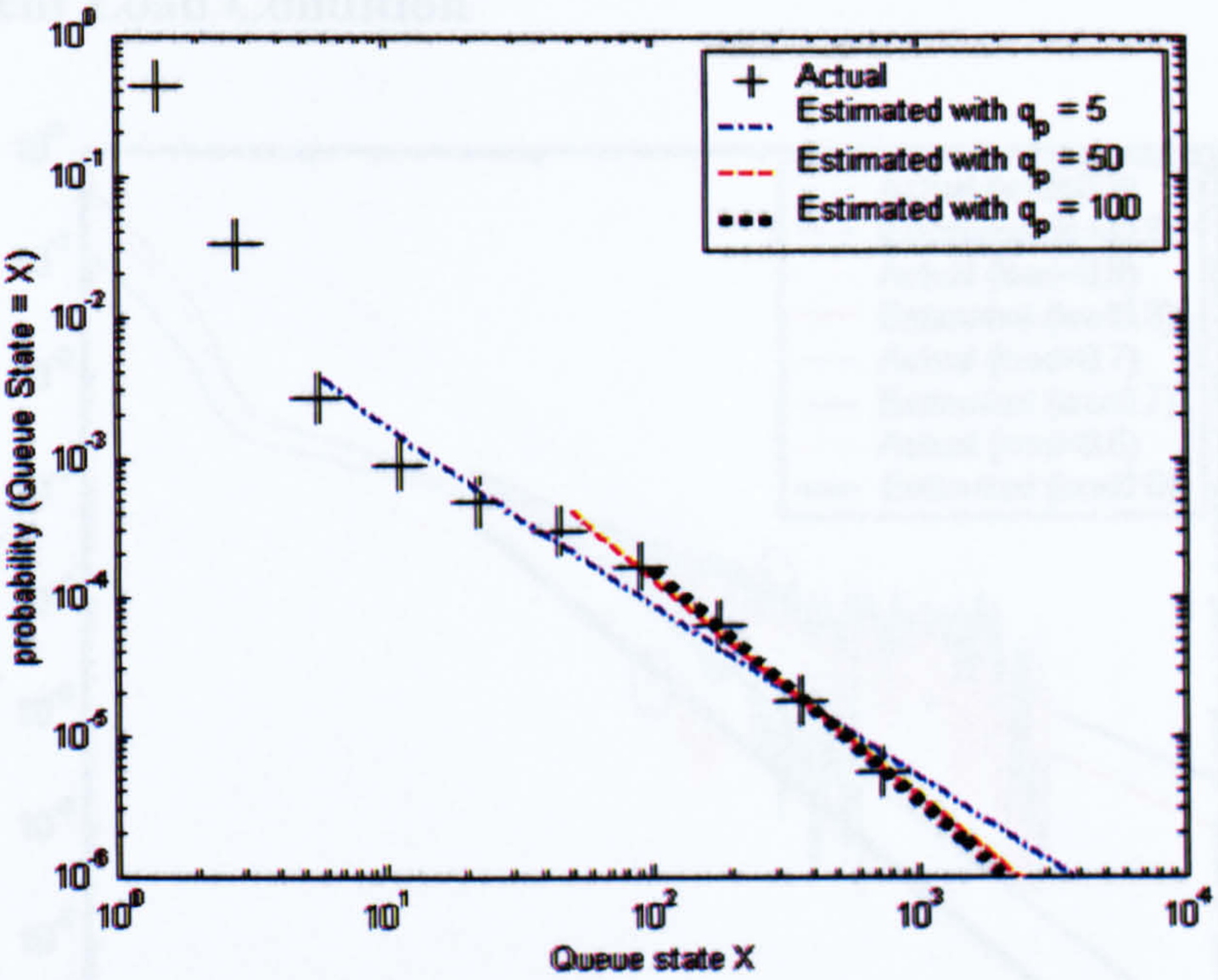


Figure 5-10 Queue length distribution in the presence of Power-law traffic with different partition point

Figure 5-10 shows that the tail of queue length distribution in the presence of Power-law traffic approximately forms a straight line in a log-log plot [PRU95a] [MA02]. It is noted that the actual queue length distribution is binned and the average value within the bin is plotted with the same rationale as discussed in Section 5.3.4.1.[15] Regarding the selection of partition point, this demonstrates similar phenomena as in the case of Markovian traffic. Both the estimates, obtained with partition points = 50 and 100, closely match with the actual queue length distribution. However, the estimate with partition point = 5 deviates from the actual one. This small value partition point falls into the packet-scale region and so gives rise to a poorer estimation result. A large partition point guarantees that it is located in the burst-scale region, but the frequency of hitting $queue_{high}$ region becomes smaller. With the same rationale as discussed in Section 5.3.1, a reasonable value for the partition point should be smaller than 100 but larger than 20.

[15] Unless otherwise stated, the plots for the distribution displayed in discontinous fashion are binned and averaged to reduce the variance and obtain better visualisation in this thesis.

5.4.2 Different Load Condition

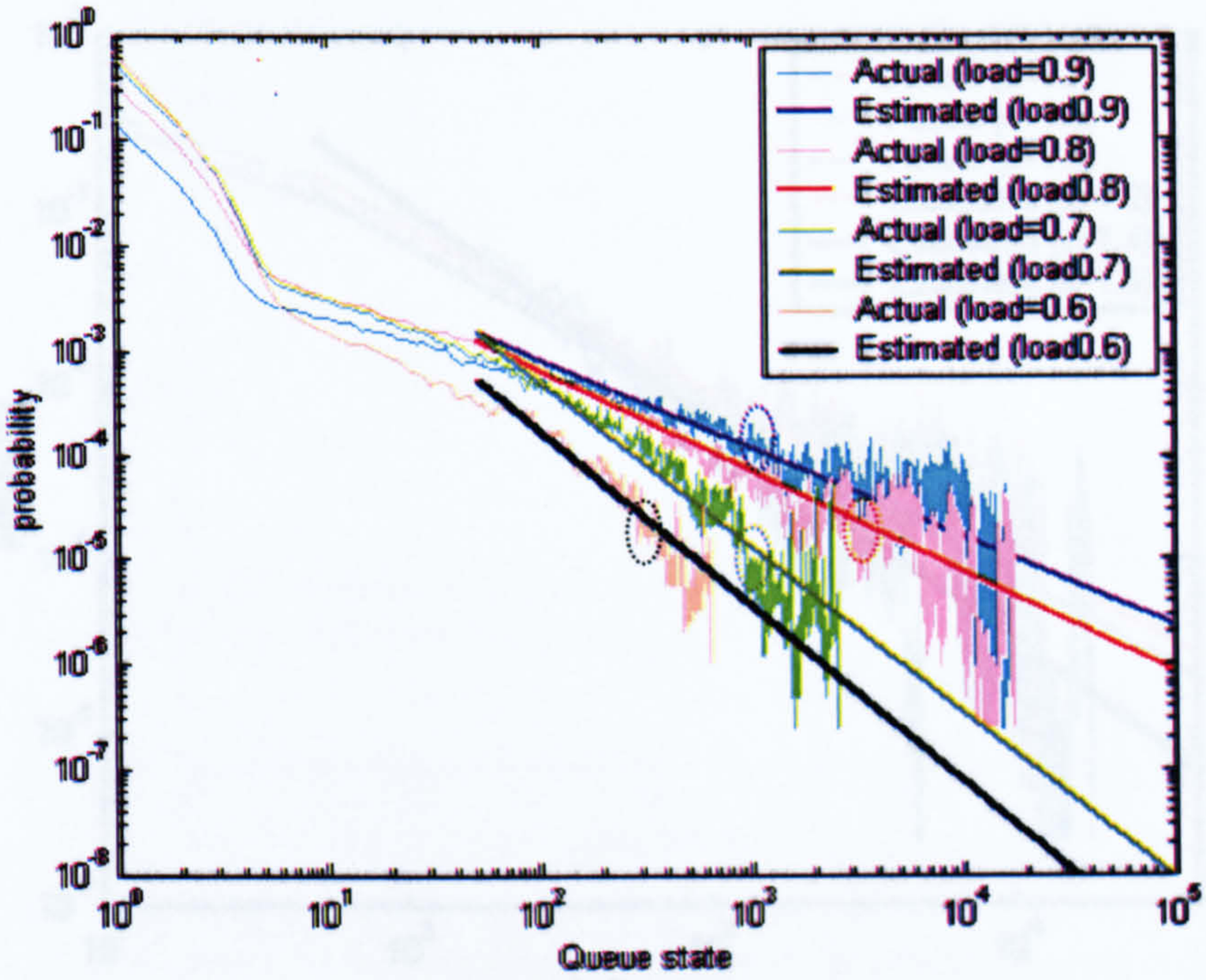


Figure 5-11 Actual queue length distribution and estimated with different load condition

Figure 5-11 shows the comparison result between the actual queue length distributions with the estimated one from our QL scheme under different load conditions. The experimental set-up is the same as that in Table 5.2. The service rate is adjusted to achieve different loads.

With reference to Figure 5-11, this shows that, again, longer queue tails happen for higher load conditions. The tail can be approximated by a straight line on a log-log scale. Hence, the queue tail exhibits power-law decay instead of exponential decay in Markovian traffic case (see Figure 5-3).

We set the partition point = 50 to perform QL Measurement. The estimated queue length distributions are as shown in Figure 5-11. Based on this experimental result, it can be seen that the estimates agree with the actual queue length distributions under different load conditions.

5.4.3 Different Shape Parameters

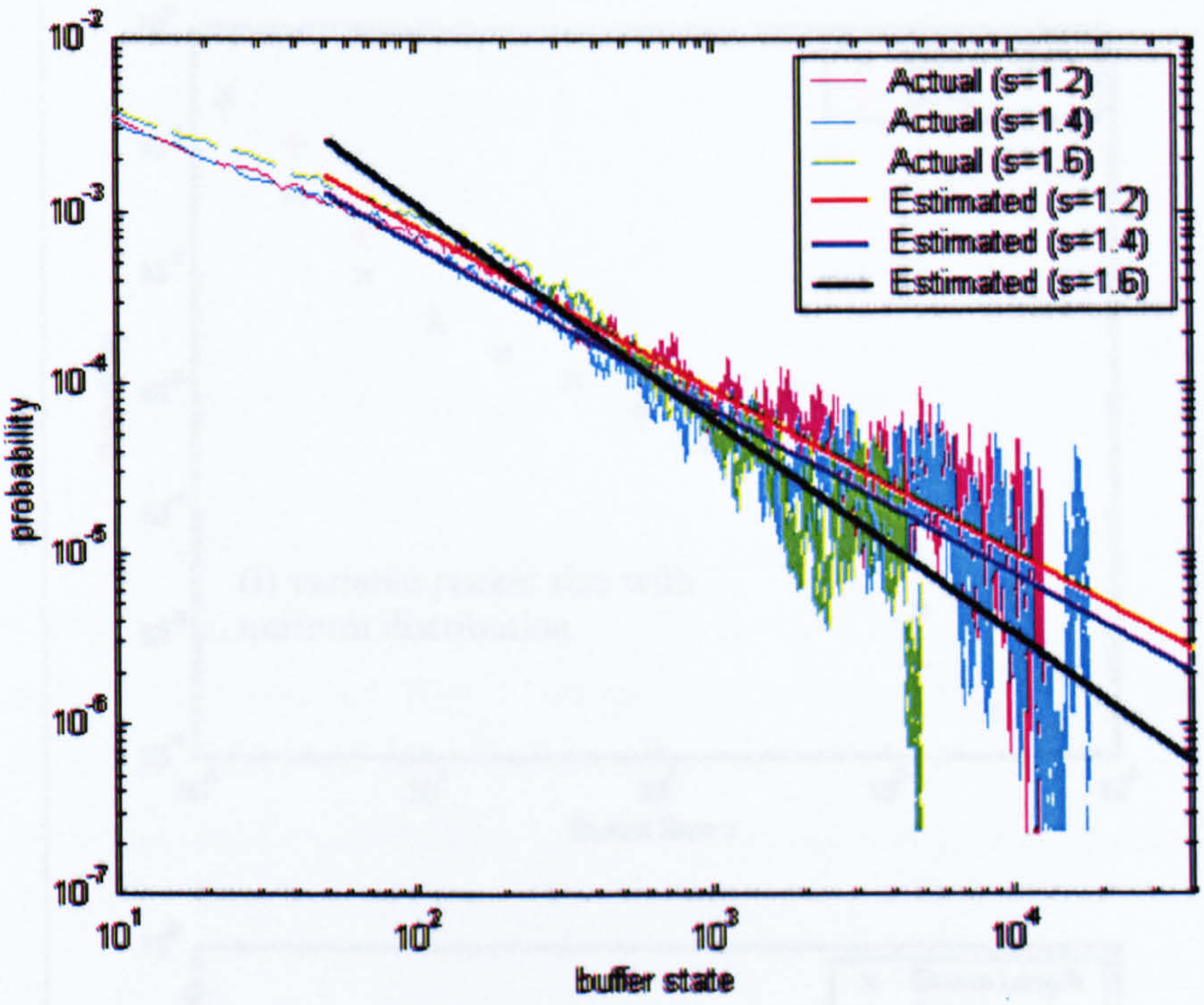


Figure 5-12 Simulations with different shape parameters

There are two parameters in the Pareto distribution: the shape parameter α, and the location parameter b (see Equation 4-1). As discussed in Section 4.1.1.2.A, to generate self-similar traffic with multiple ON/OFF sources, the value of the shape parameter for ON and OFF period should be in the range of 1 to 2. The location parameter controls the mean value of the distribution.

Figure 5-12 shows simulation results with different shape parameters. The experimental set-up is shown in Table 5.2. The load is 0.8. The smaller the shape parameter, the higher the probability of an extreme value can be obtained in the Pareto distribution and so the more bursty the traffic. This is illustrated in Figure 5-12. The queue tail decays faster for the case of shape parameter = 1.6, whereas, a flatter queue tail occurs for the case of shape parameter = 1.2.

The queue length distribution is estimated with QL Measurement with partition point = 50. Again, the estimated queue length distributions closely match with the actual distributions.

5.4.4 Variable Packet Size

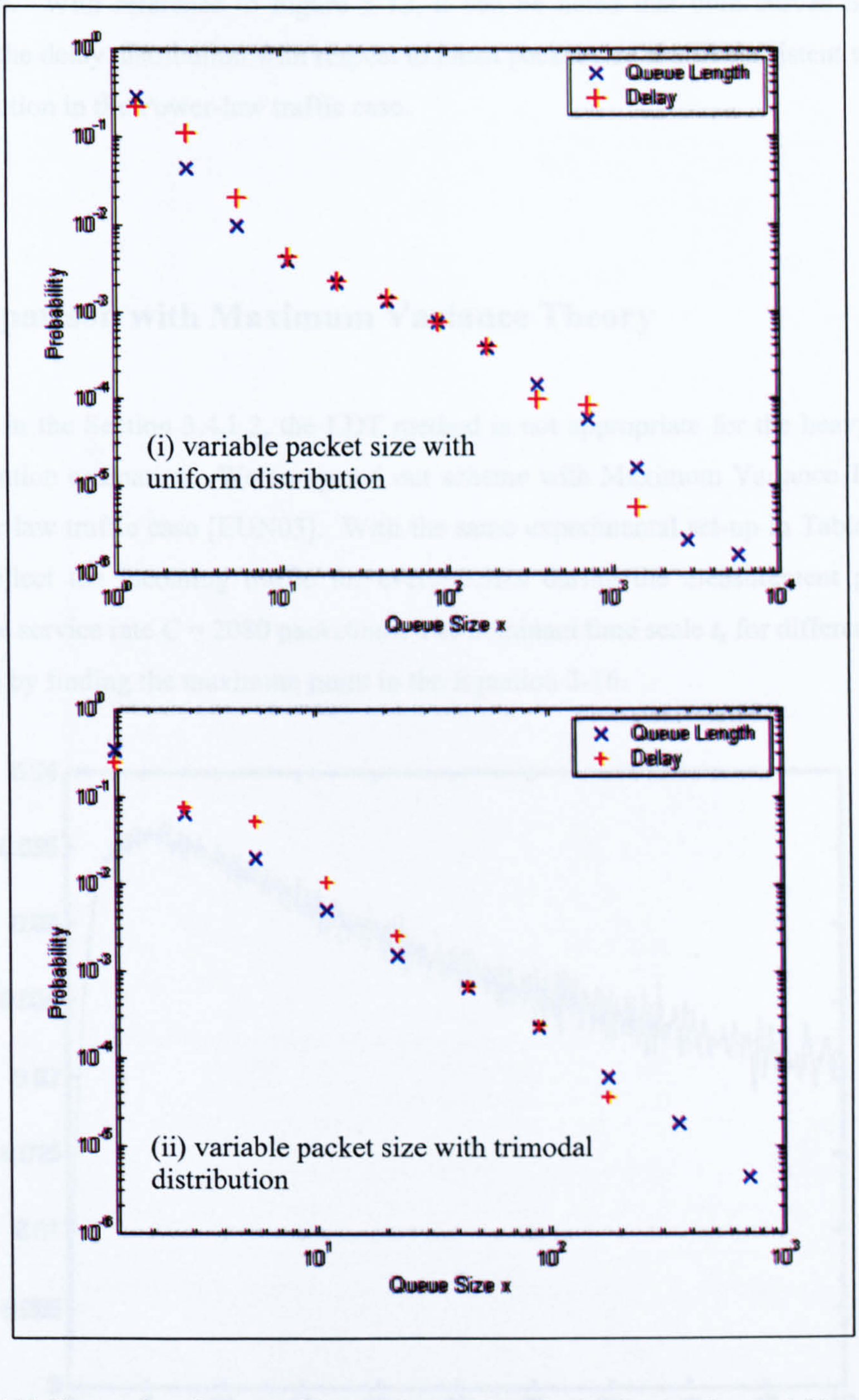


Figure 5-13 Comparison between delay and queue length distribution for variable packet size

We repeated the experiment as in Section 5.3.4. The packet size has a uniform distribution or a trimodal distribution[16]. The load condition is 0.6. The foreground traffic is of interest, and it is multiplexed with 100 background sources. The parameters of the ON/OFF traffic remain the same as in Table 5.2 but with variable packet size. We collect the data for the measurement period = 3600

[16] Please refer to Appendix D for the simulations of variable packet size with bimodal distribution.

sec and compare the queue length distribution obtained from the foreground traffic's delay data with the actual one. With reference to Figure 5-13, it can be noted that both curves overlap, which suggests that the delay distribution with respect to mean packet size is still consistent with the queue length distribution in the Power-law traffic case.

5.4.5 Comparison with Maximum Variance Theory

As discussed in the Section 3.4.1.2, the LDT method is not appropriate for the heavy-tailed queue length distribution estimation. We compared our scheme with Maximum Variance Theory (MVT) for the Power-law traffic case [EUN03]. With the same experimental set-up in Table 5.2, to apply MVT, we collect the incoming traffic for every 0.5ms during the measurement period (=3600 seconds). The service rate $C = 2080$ packet/sec. The dominant time scale t_x for different queue size x is determined by finding the maximum point in the Equation 3-16.

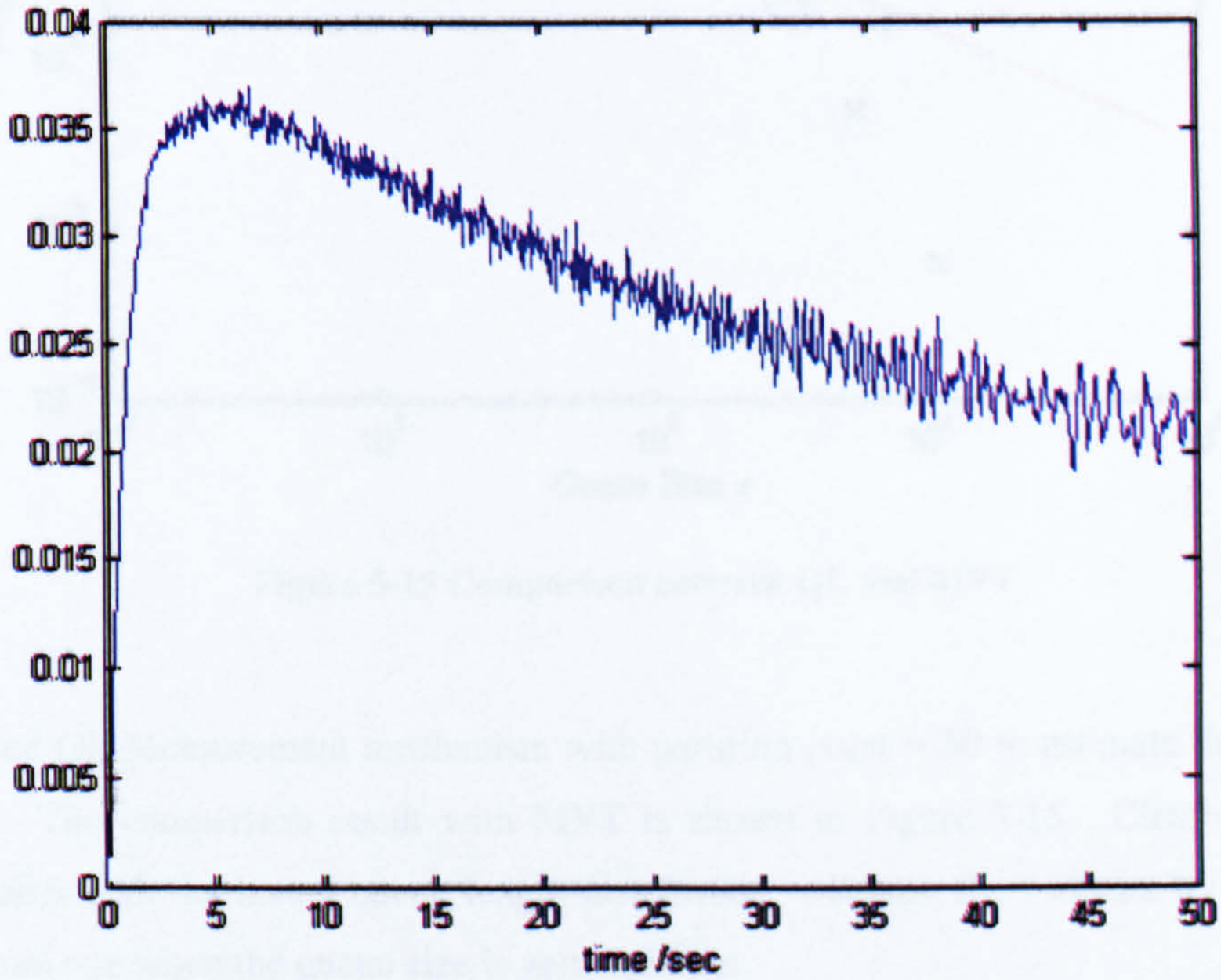


Figure 5-14 Plot of $v_t/(x-\kappa_t)^2$ versus time with queue size x =1120

Figure 5-14 shows the plot of $v_t/(x-\kappa_t)^2$ versus time t with respect to a particular queue size $x = 1120$. It can be seen that the curve reaches the maximum point at $t = 5.9$ second. Therefore, the dominant time scale (dts) for queue size $x = 1120$ is 5.9 second. Based on this dominant time scale, we obtain

the value κ_t = -5416 and $v_t = 1.53 \cdot 10^6$. The queue length distribution at the queue size x can be determined by substituting the value x, κ_t and v_t into Equation 3-18 which gives $3.83 \cdot 10^{-9}$.

Likewise, we estimate the queue length distribution $Q(x)$ at various points x as follows:

x	35	70	140	280	560	1120
dts /sec	0.4	0.6	0.85	1.4	2.4	5.9
$Q(x)$	$2.4 \cdot 10^{-4}$	$9.02 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$	$3.6 \cdot 10^{-6}$	$4.36 \cdot 10^{-7}$	$3.83 \cdot 10^{-9}$

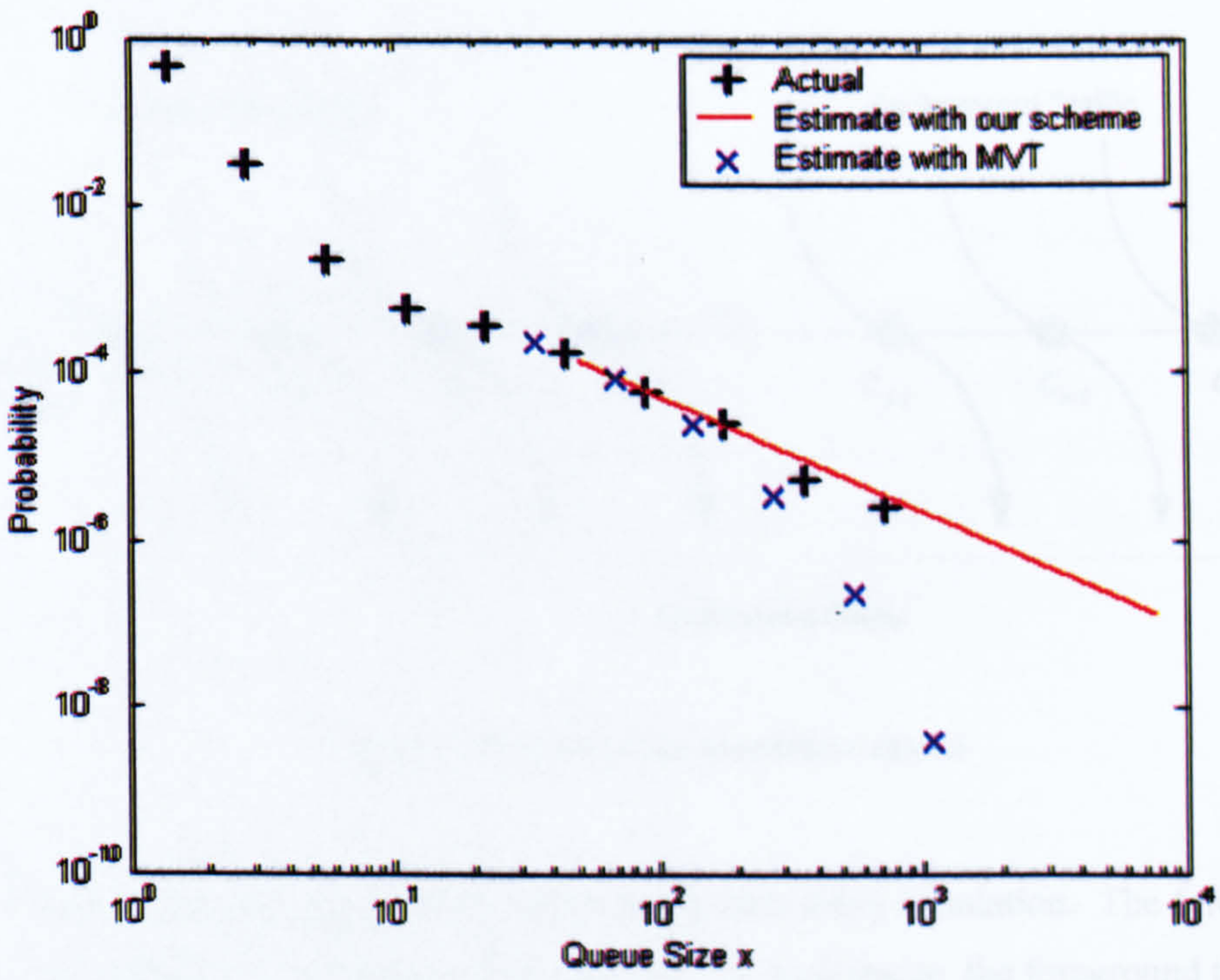


Figure 5-15 Comparison between QL and MVT

We performed QL Measurement mechanism with partition point = 50 to estimate the queue length distribution. The comparison result with MVT is shown in Figure 5-15. Clearly, our estimate closely matches with the actual queue length distribution, whereas, the estimate by MVT deviates from the actual one when the queue size is getting larger.

5.5 End-to-end Delay Performance Simulation

In the previous section, we examined the effectiveness of the QL Measurement scheme in estimating the per-hop queue length distribution. The per-hop queue length distributions along a path are used to determine the end-to-end delay distribution as explained in Section 4.3[17].

In Section 4.3, we discussed the role of the foreground traffic and background traffic. The foreground traffic is of interest and this will be interfered with by the background traffic in the end-to-end simulation model.

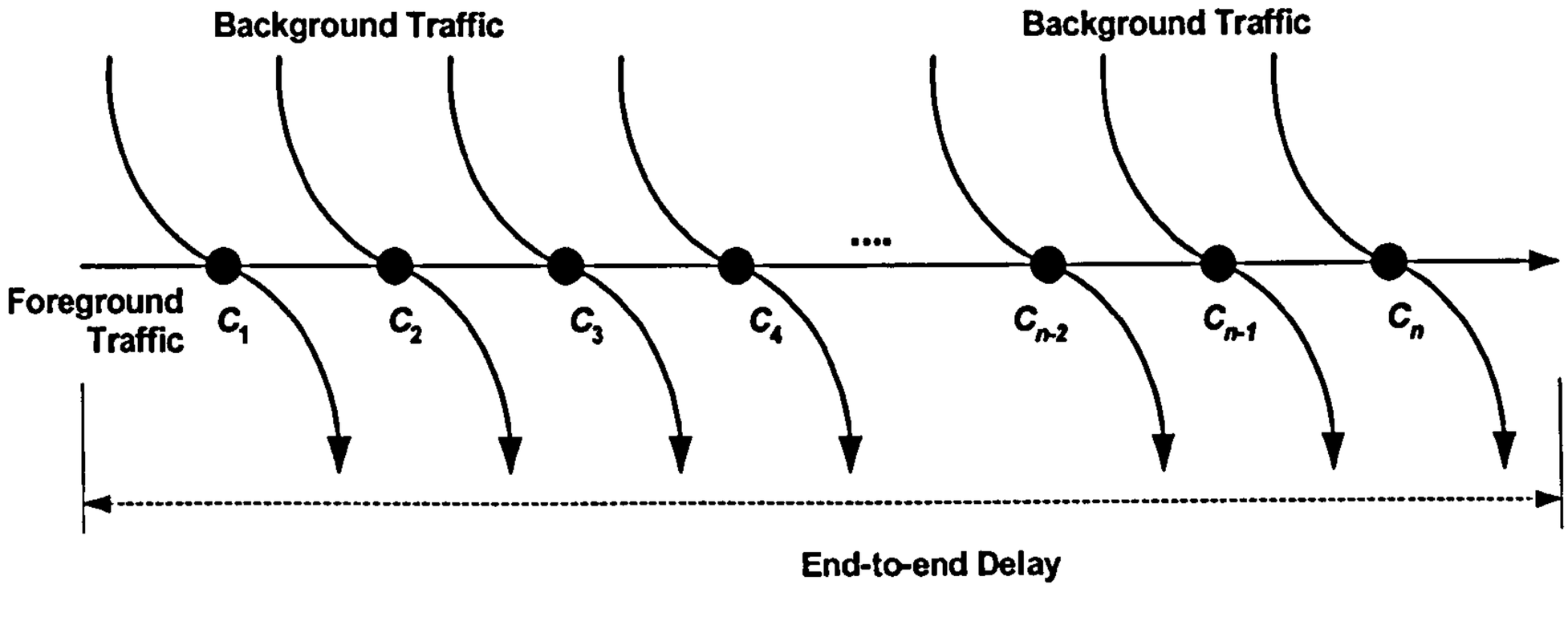


Figure 5-16 End-to-end simulation model

Figure 5-16 depicts the simulation set-up for an end-to-end delay simulation. The foreground traffic traverses *n* queues before reaching the receiving end. At each queue, the foreground traffic is mixed with the background traffic. The background traffic will leave the system after multiplexing at the queue as discussed in Section 4.3. This model is widely used to study the end-to-end delay performance [SEU00] [LIU02] [STE02].

We estimate the end-to-end delay distribution by convolving the per-hop queue length distributions obtained by the QL Measurement scheme. This estimate is compared to the actual end-to-end delay distribution which is obtained by the delay data of every foreground traffic packet.

---

[17] We assume NOC is capable of awaring of the path the foreground traffic traverses, like ATM or MPLS

5.5.1 An Issue of Path Properties

We have assumed that all the packets of the foreground traffic follow the same route during measurement. This assumption is held in connection-oriented networks like ATM. However, it may not be true in the regime of connectionless-oriented networks like the Internet. Different routes may be taken from packet to packet. Several experiments have been performed to study route persistence in the Internet [FEI98] [ZHA00]. In [ZHA00], the results reveal that "the routes very often persist for at least a day, namely that most paths are persistent over time scales of many hours to days, but a fair number of paths are persistent only for quite shorter time scales (but still up to several hours)". Based on these observations, it is fair enough to assume that the route does not change in the time scale of an hour.

Regarding the hop count experienced by the foreground traffic, as shown in Figure 5-16, in [RIB03], congestion largely occurs at the edge of the network close to the sending or receiving end. Thus data packets are likely to encounter two congested queues, one at each end of their paths. As a consequence, the scenario of two congested hops is usually considered. Taking the Internet as a reference, the longest route recorded in the Internet study [FEI98] has a longest hop-count of 27. Considering the number of hops involved in the backbone network, it is sufficient to simulate the end-to-end path with hop count from 2 to 10.

5.5.2 End-to-end Delay Performance Simulation (Markovian Traffic)

In this experiment, the foreground traffic needs to traverse 3, 5 and 10 hops respectively before reaching the receiving end. At each queue (node), there are 99 background sources multiplexed into the queue. The traffic characteristics for both foreground and background are the same as shown in Table 5.1. The load at odd numbered queues 1,3,5,7,9 is set to 0.7, i.e. the service rate is 8490 packets/sec, while the load at even numbered queues 2,4,6,8,10 is equal to 0.9 with the service rate 6722 packets/sec. We perform QL Measurement for the observation period =3600 sec with the partition point = 50 to collect the necessary data.

Q_k	$\bar{q}_{low}$	$\bar{q}_{high}$	p_{low}	p_{high}	$\hat{c}_{bmk}$	$\hat{\eta}_{bmk}$
1	1	97.3337	0.9992	7.5273e-004	4.7251e-005	0.9789
2	4	686.2734	0.6110	0.3890	6.6248e-004	0.9984
3	1	78.9548	0.9994	5.6934e-004	1.1807e-004	0.9655
4	5	600.6859	0.6203	0.3797	7.5653e-004	0.9982
5	1	192.3838	0.9986	0.0014	1.3782e-005	0.9930
6	5	673.0841	0.5985	0.4015	6.9930e-004	0.9984
7	1	140.3441	0.9988	0.0012	2.3249e-005	0.9889
8	4	635.4143	0.6232	0.3768	7.0220e-004	0.9983
9	1	130.1702	0.9992	7.5531e-004	1.7870e-005	0.9875
10	4	612.0734	0.6191	0.3809	7.4219e-004	0.9982

Table 5-3 Measurement data and the estimated result (10 hops)

Table 5.3 shows the measurement data and the estimated burst-scale decay rate and decay constant estimated by using Equation 4-15 and Equation 4-17. We set the reference service rate as 6722 packet/sec and so the queue state distribution at odd-numbered queue 1,3,5,7,9 needs to be normalised as discussed in Section 4.4. Then the end-to-end delay distribution is obtained by convolving every queue state probability distribution along the path. We collect all the foreground traffic packets and record the end-to-end delay time i.e. the time difference between leaving the sending and reaching the receiving end. To make the comparison, the data of the packet delay is converted in terms of queue size with respect to the reference service rate as shown in Equation 4-26.

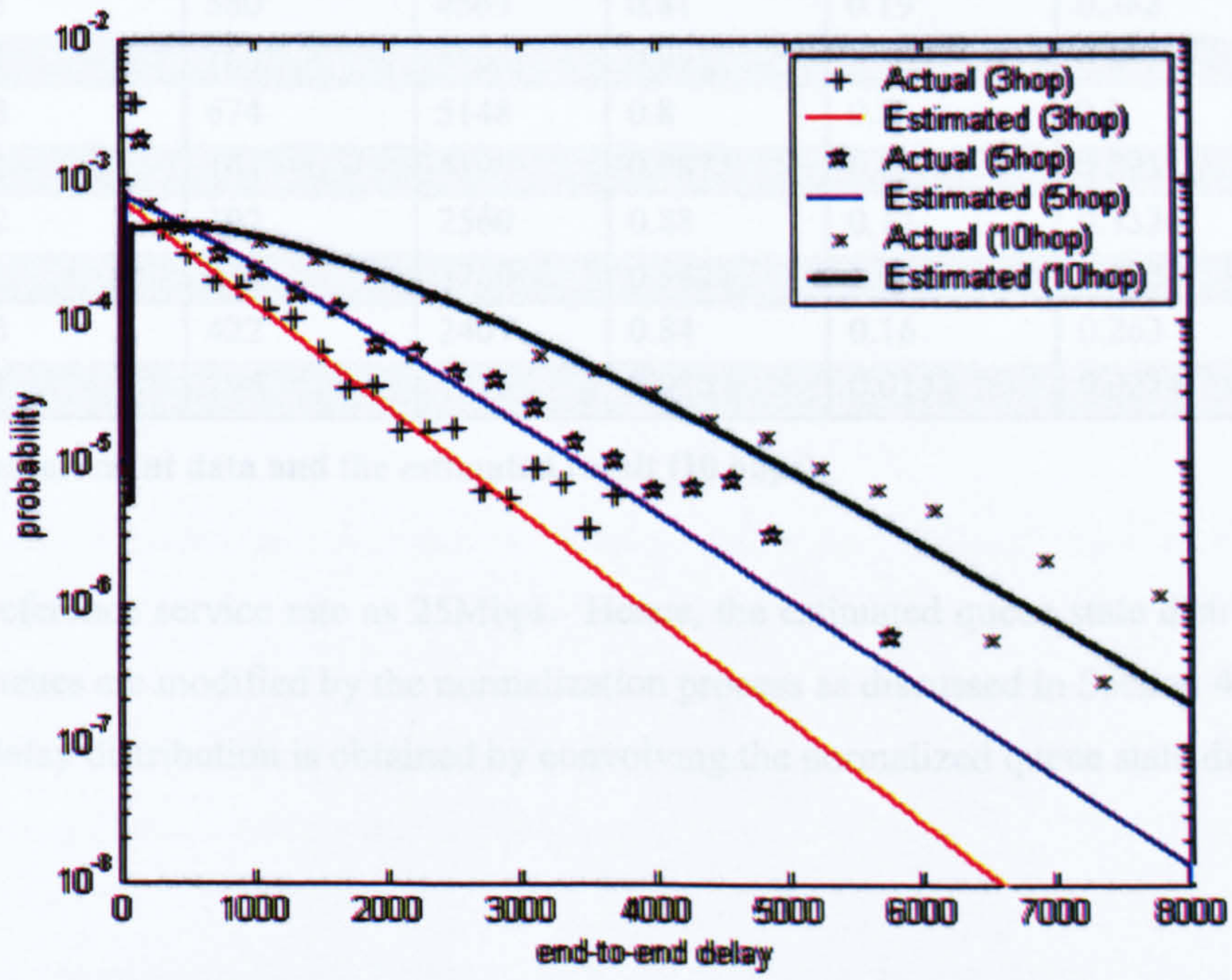


Figure 5-17 End-to-end delay distribution

Figure 5-17 shows the comparison between the experimental results and the estimation result from the measurement data. It can be noted that the estimated end-to-end delay distribution has a good agreement with the experimental result. The tail of the estimated end-to-end delay distribution can be used for the verification of the SLA end-to-end delay performance. This will be discussed in Section 7.

5.5.3 End-to-end Delay Performance Simulation (Power-law Traffic)

We repeated the experiment above, but this time for Power-law traffic instead. Again, the foreground traffic traverses n hops (n=3, 5 and 10), and 100 background sources are multiplexed at each queue. Both foreground and background traffic share the same characteristics. Mean ON time = 1 sec, Mean OFF time = 10 sec, with Pareto distribution and the shape parameter = 1.4. The service rate for odd numbered nodes is 20Mbps, while that of even numbered nodes is 25Mbps. Table 5.3 shows the measurement results. The decay rate and the decay constant are determined using Equation 4-23 and Equation 4-24.

Q_k	$\bar{q}_{low}$	$\bar{q}_{high}$	B	p_{low}	p_{high}	$\hat{c}_{bpk}$	$\hat{\eta}_{bpk}$
1	2	329.6	1859	0.87	0.13	0.38	-1.425
2	1	122	448	0.99	0.01	0.57	-2.01
3	3	580	4563	0.81	0.19	0.382	-1.375
4	1	168.7	810	0.991	0.009	0.192	-1.8
5	3	674	5148	0.8	0.2	0.3	-1.325
6	1	141	419	0.9875	0.0125	0.095	-1.57
7	2	392	2560	0.88	0.12	0.3326	-1.425
8	1	339	1760	0.9847	0.0153	0.0324	-1.367
9	3	422	2407	0.84	0.16	0.263	-1.331
10	1	296	1225	0.986	0.014	0.0224	-1.3

Table 5-4 Measurement data and the estimated result (10 hops)

We set the reference service rate as 25Mbps. Hence, the estimated queue state distributions at odd numbered queues are modified by the normalization process as discussed in Section 4.4.2. Then, the end-to-end delay distribution is obtained by convolving the normalized queue state distributions.

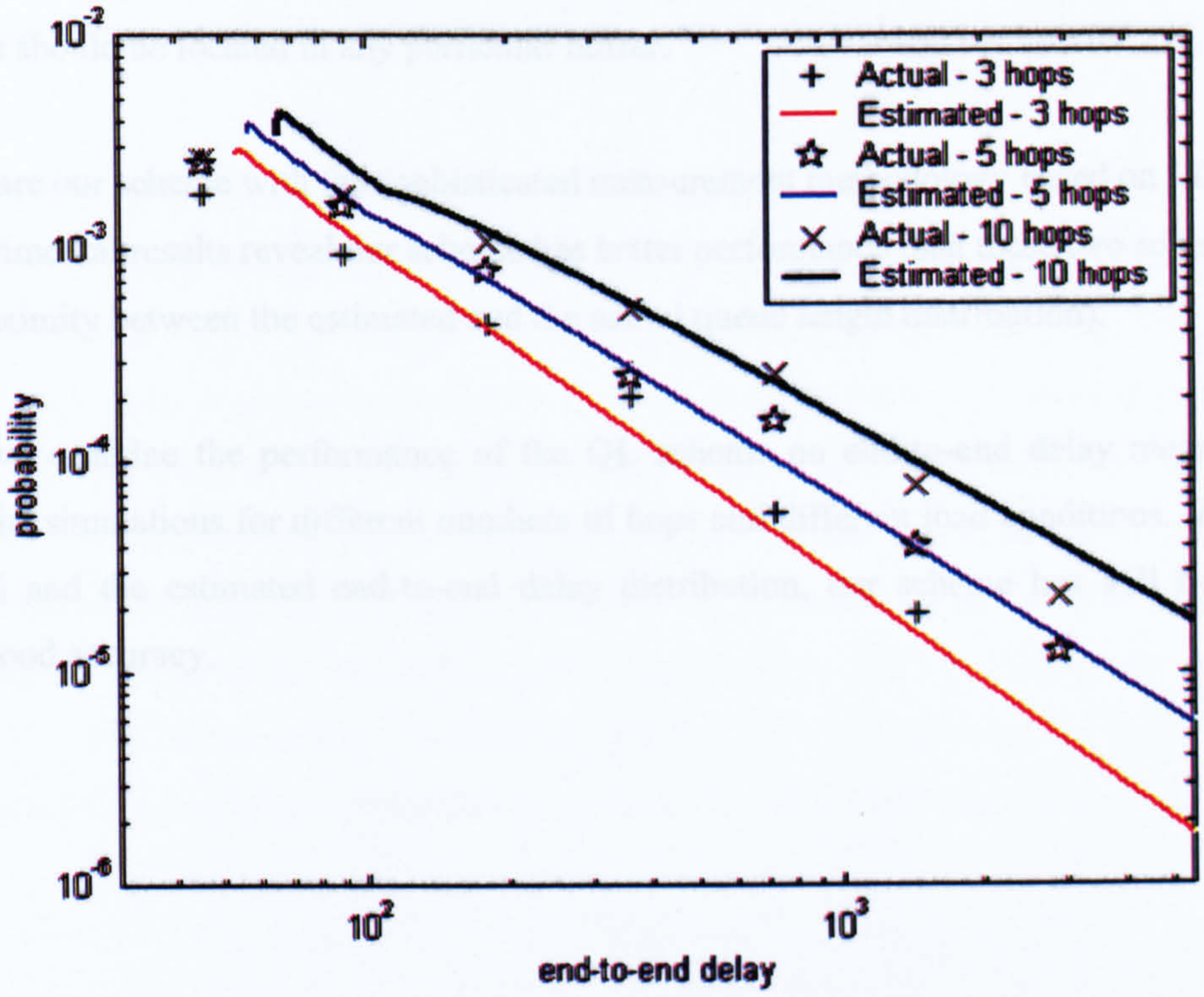


Figure 5-18 End-to-end delay distribution

Figure 5-18 depicts the comparison between the estimated and the actual end-to-end delay distribution for number of hops =3, 5 and 10. It can be seen that there is a good agreement between the two distributions. This experiment shows that our QL methodology is even able to cope with packet traffic which is governed by extremely high variance power-law distributions.

5.6 Summary

In this chapter, we assess the effectiveness of our proposed measurement scheme QL Measurement from various aspects. First of all, we study the queue length distribution under various conditions: (1) Traffic type (Markovian or Power-law traffic), (2) different load conditions, (3) different degrees of traffic burstiness. Two types of queue tail distributions, exponential decay (Markovian traffic) or power-law decay (Power-law traffic) are verified.

We investigate the impact of the partition point. Experimental results show that the per-hop queue length distribution estimation result is insensitive to the partition point provided that it is located in

the burst-scale region. This is a particular advantage, since it cannot be easily known in advance where this should be located in any particular buffer.

We compare our scheme with the sophisticated measurement methodology based on LDT and MVT. The experimental results reveal our scheme has better performance than these two schemes (in terms of the proximity between the estimated and the actual queue length distribution).

Finally, we examine the performance of the QL scheme on end-to-end delay measurement. We perform the simulations for different numbers of hops and different load conditions. By comparing the actual and the estimated end-to-end delay distribution, our scheme has still been shown to provide good accuracy.

Chapter 6 The Effect of Bandwidth/Link Sharing

In this chapter, we study the effect of the buffer scheduling mechanism on our QL Measurement scheme. Scheduling mechanisms are commonly employed to allocate the link bandwidth to different classes of traffic, for example Class-Based Queuing, or Diffserv [BLA98]. For any work conserving scheduler, the spare bandwidth will be given to the other non-empty queues. This is also known as the bandwidth stealing effect [BRE00a][KUZ01]. Due to the bandwidth stealing effect, it is inappropriate to make any prior assumption on the received bandwidth. Our end-to-end delay measurement relies on the knowledge of the bandwidth of each local queue. Therefore, we develop a measurement methodology called Bandwidth Measurement (BW) to measure the actual received bandwidth. The measurement result obtained by BW Measurement is input into the QL Measurement scheme for the normalization process. This is finally evaluated through simulation.

6.1 Queuing Discipline

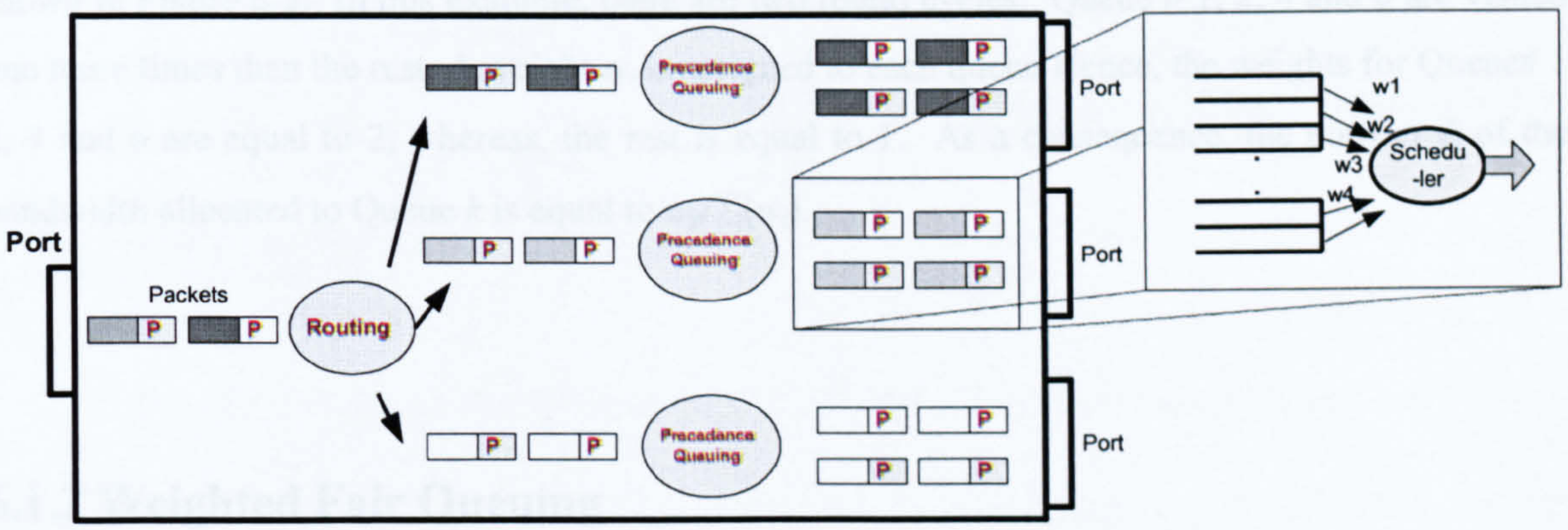


Figure 6-1 Classification and queuing discipline

Figure 6-1 depicts the general architecture of a router. The incoming packet is routed / switched into the corresponding output links' dispatcher. The dispatcher checks the precedence content in the packet. Based on this information, the packet will be en-queued into the corresponding queue. The link bandwidth is allocated into each queue (traffic class) by a scheduler. Common scheduler mechanisms are Weighted Round Robin (WRR) or Weighted Fair Queuing (WFQ) [PAR93]. Many other derivations exist, mainly differing in the degree of complexity for a hardware implementation: virtual time [PAR93], self-clocked fair queuing [GOL94] and virtual spacing [ROB94]. In this chapter, only WRR and WFQ are considered.

6.1.1 Scheduler Mechanism (Weighted Round Robin)

The basic idea of WRR scheduling is described as follows: All queues are visited a different number of times during N cycles. If certain queues are visited more times than others, then they will get a relatively high share of the bandwidth (see Figure 6-2).

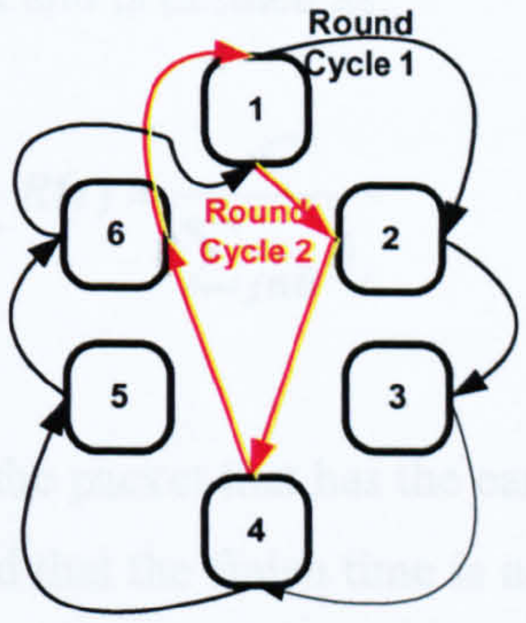


Figure 6-2 Weighted Round-Robin Multiplexing

[KAT91] has shown a scheduling algorithm based on the concept of weighted circular scan as shown in Figure 6-2. In this example, there are two round cycles. Queue # 1, 2, 4 and 6 are visited one more times than the rest. A weight w_i is assigned to each queue Hence, the weights for Queue# 1, 2, 4 and 6 are equal to 2, whereas, the rest is equal to 1. As a consequence, the portion ϕ_i of the bandwidth allocated to Queue k is equal to $w_k/\sum(w_i)$.

6.1.2 Weighted Fair Queuing

WFQ supports the fair distribution of bandwidth for *variable-length* packets by approximating a generalized processor sharing system. While generalized processor sharing system is a *theoretical* scheduler that cannot be implemented, its behaviour is similar to a weighted bit-by-bit round-robin scheduling discipline. In WFQ, similar to WRR, each queue is assigned a weight w_i and so the portion of bandwidth $\phi_i = w_i/\sum w_i$ will be allocated to the queue i. WFQ takes the spare bandwidth into consideration and distributes this spare bandwidth to all non-empty queues by making use of a monitoring parameter called Round Number $R(t)$. The operating mechanism of WFQ is as follows. Each arriving packet is given a virtual finish time. The virtual finish time of packet k at queue i denoted as $F(k,i)$ is computed as follows:

$$F(k,i) = \max\{F(k\text{-}1,\ i),\ R\{a(k,i)\} + L(k,i)/\phi_i \qquad \textbf{Equation 6-1}$$

with $F(0,i) = 0$, $a(k,i)$ and $L(k,i)$ are the arrival time and the size of the packet respectively. $R(t)$ is the virtual round number function representing the progression of virtual time in the simulated generalized processor sharing model and is defined as:

$$\frac{d}{dt}R(t) = \frac{C}{\sum_{j \in B}\phi_j} \qquad \textbf{Equation 6-2}$$

The scheduler selects and forwards the packet that has the earliest (smallest) finish time from among all of the queued packets. It is noted that the finish time is not the actual transmission time for each packet. Instead, the finish time is a number assigned to each packet that represents the order in which packets should be transmitted on the output port as shown in Figure 6-3.

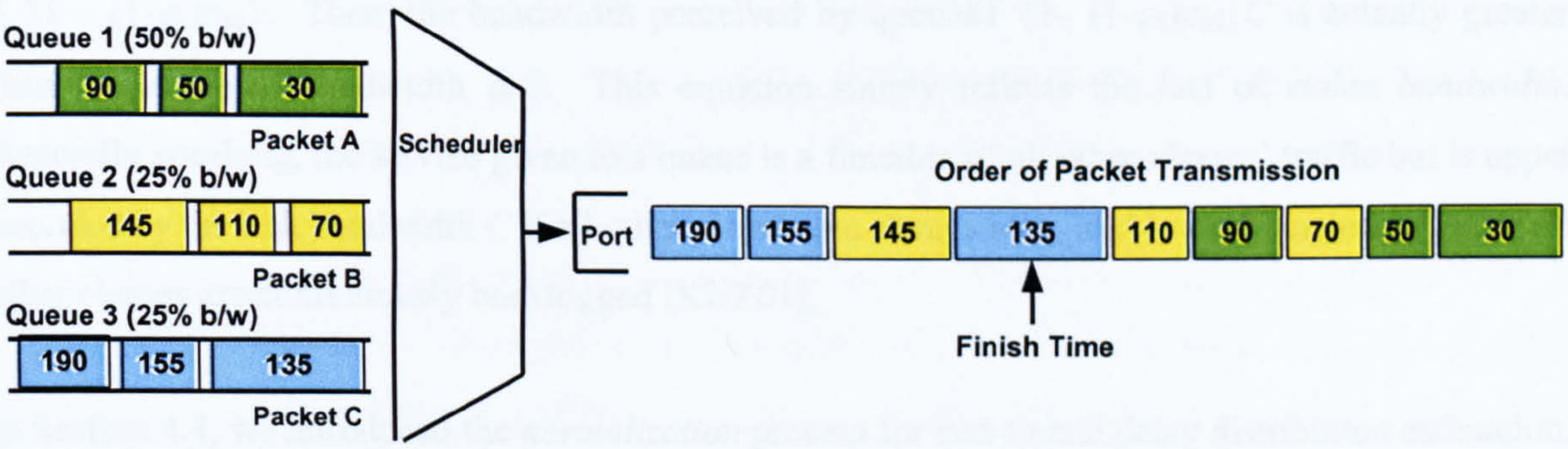


Figure 6-3 Weighted Fair Queuing (WFQ) – Service according to packet Finish Time

6.2 Bandwidth Stealing

With a work conserving scheme like WRR or WFQ, the server is always busy when there are non-empty queues in the system. The unused bandwidth of all empty queues will be shared by the non-empty queues. In other words, the unused bandwidth is *stolen* by other non-empty queues. This effect was also addressed in the works [BRE00a] [KUZ01].

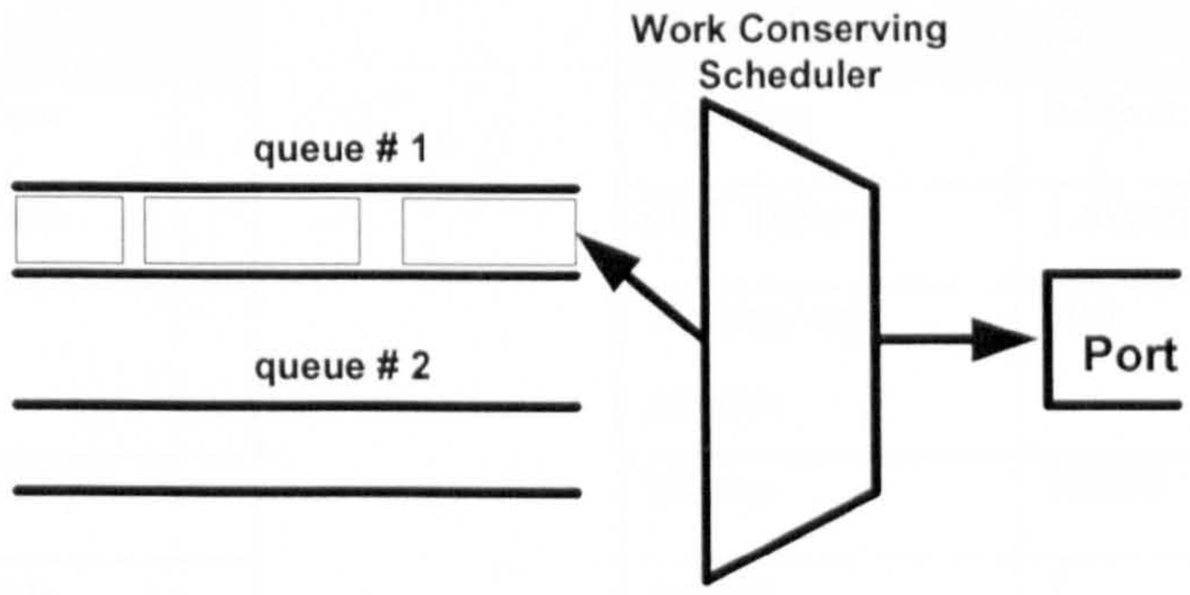


Figure 6-4 Bandwidth stealing

In Figure 6-4, there are two classes of traffic. The queues are assigned with the weight ϕ_1 and ϕ_2. Let's assume p_{b1} and p_{b2} represent the probability of queue#1 and queue#2 being non-empty respectively. When both queues are not empty, queue#1 will be served at a rate $\phi_1 C$. However, when queue#1 is non-empty but queue#2 is not, then queue#1 will receive full bandwidth. Therefore, the average bandwidth queue#1 receives is given as $p_{b2}\phi_1 C + (1\text{-}p_{b2})C$, i.e. $\{1 - (1\text{-}\phi_1)p_{b2}\}C$. Since $\phi_1 \leq 1$, and $(1\text{-}p_{b2}) > 0$ for a stable system, therefore, $(1\text{-}p_{b2})\phi_1 \leq (1\text{-}p_{b2})$. Clearly, $\phi_1 \leq \{1 - (1\text{-}\phi_1)p_{b2}\}$. Then, the bandwidth perceived by queue#1 $\{1- (1\text{-}\phi_1)p_{b2}\}C$ is actually greater than the allocated bandwidth $\phi_1 C$. This equation simply reflects the fact of *stolen bandwidth*. Generally speaking, the service given to a queue is a function of all other classes' traffic but is upper bounded by the link bandwidth C if all other classes are always idle, and lower bounded by $\phi_i C$ if all other classes are continuously backlogged [KUZ01].

In Section 4.4, we introduced the *normalization* process for end-to-end delay distribution estimation. This process requires the knowledge of the service rates received by each sub-queue along the path. Due to the *bandwidth stealing* effect, it is not appropriate to assume the received bandwidth simply as the product of the assigned weight of the scheduler with the link bandwidth. This is illustrated by the following examples.

(A) Traffic class #1	
On Time	0.96sec
OFF Time	1.69sec
Number of sources	100
bit rate	70808bps
service rate	3Mbps

Table 6-1 (A) Single Traffic Class Setup

(B) Traffic class	#1	#2
On Time	0.96sec	0.96sec
OFF Time	1.69sec	1.69sec
Number of sources	100	30
bit rate	70808	708080
weight	1	5
service rate	18Mbps	

Table 6-1 (B) Two Traffic Classes Setup

In experiment A, there is only one traffic class. The queue multiplexes 100 Markovian traffic sources and has a service rate of 3Mbps to achieve 0.8 utilization. Figure 6-5 shows the queue length distribution. It is noted that burst-scale queuing effect occurs under this condition. In experiment B, we add one extra traffic class into the node. The traffic class #2 carries 30 ON/OFF traffic which generate packet traffic 10 times as bursty as traffic class #1 (see Table 6-1). WRR is used for bandwidth allocation. According to the weight assignment, one-sixth of the bandwidth is supposed to be given to traffic class #1 in theory, i.e. 3Mbps bandwidth is distributed to traffic class #1 which is the same as that in experiment A.

The utilization in traffic class #2 is deliberately set low for illustration purposes. The queue#2 is empty for a large portion of time. The spare bandwidth will be given to queue#1. Figure 6-6 shows the queue length distribution of queue#1. It is noted that the burst-scale queuing effect disappears, but only packet-scale queuing effect occurs that results in short queue lengths. This experiment illustrates that the actual bandwidth obtained by a queue can be much greater than the pre-assigned bandwidth when a scheduler is employed. The amount of the extra gained bandwidth depends on how busy the other queues are.

As discussed in Section 4.2, the information of the service rate is essential to relating the packet delay time with the queue size x. If we use the theoretical service rate for queue#i as $\phi_i C$, clearly it will under-estimate the actual received bandwidth. As a consequence, the queue length x based on this wrong service rate will predict a greater queuing delay compared with the actual one. This is as shown in Figure 6-7.

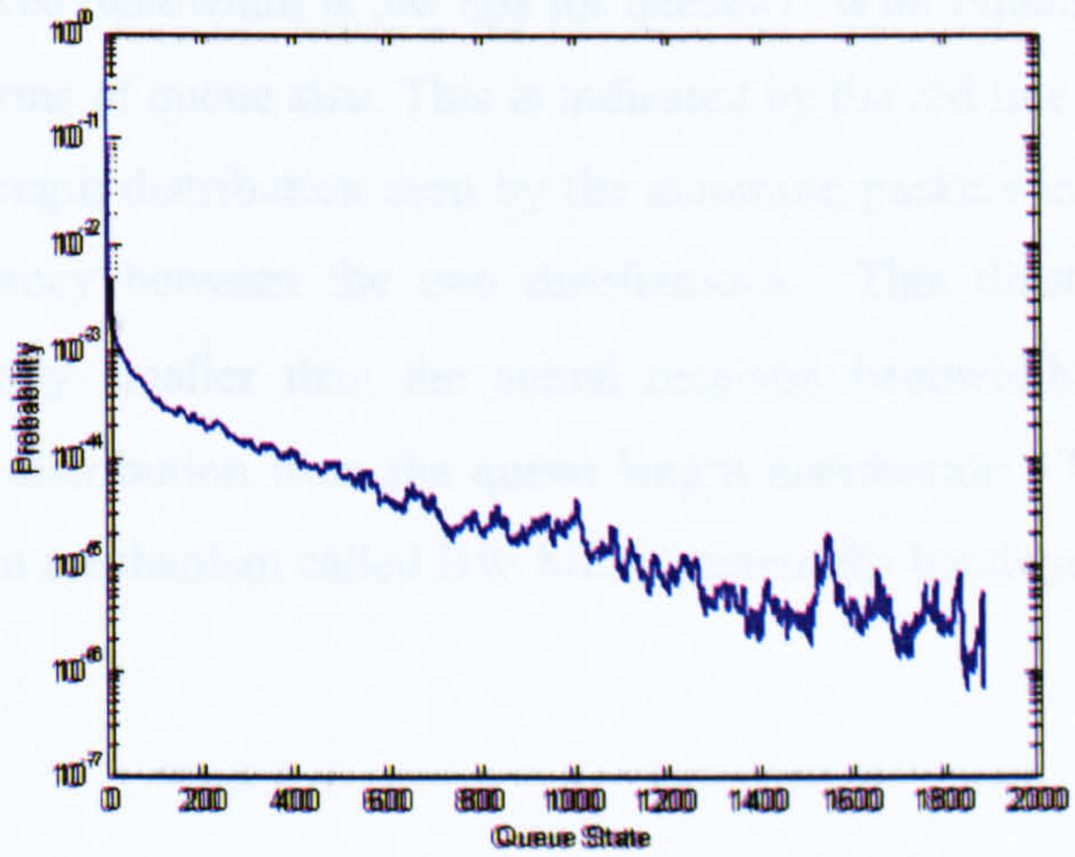


Figure 6-5 Queue length distribution for single class traffic

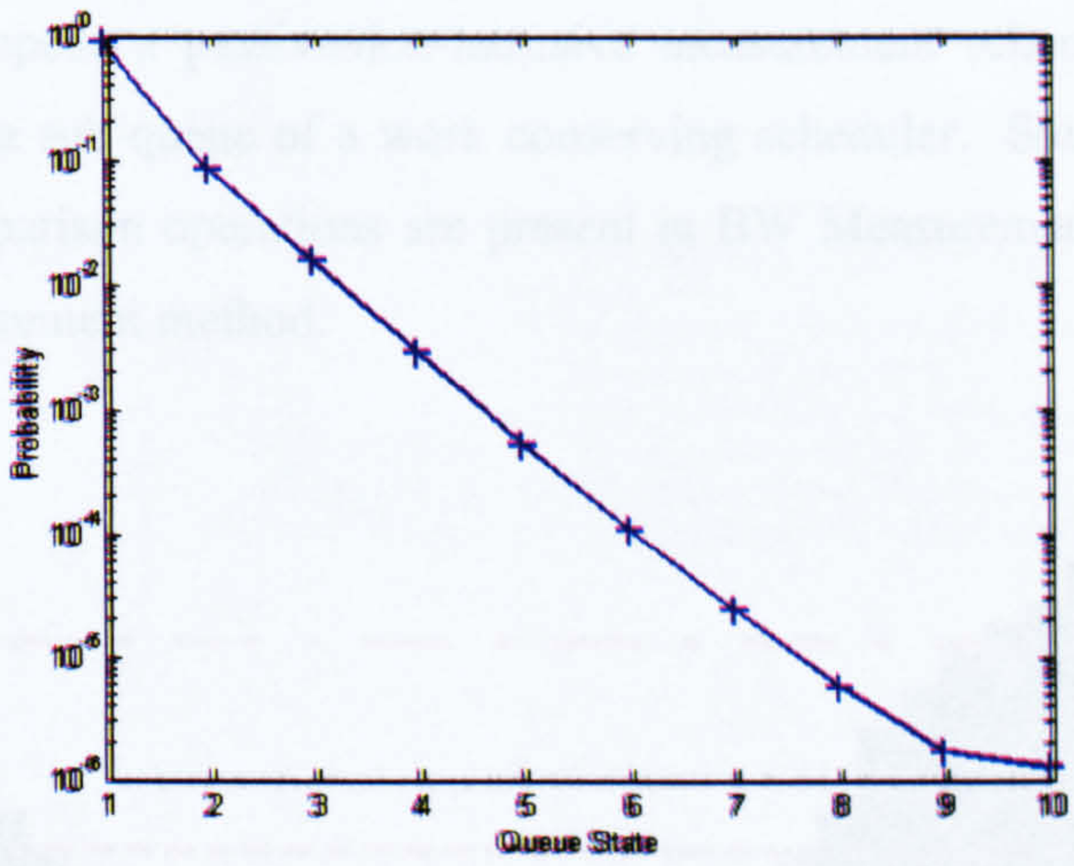


Figure 6-6 Queue length distribution for class 1 traffic

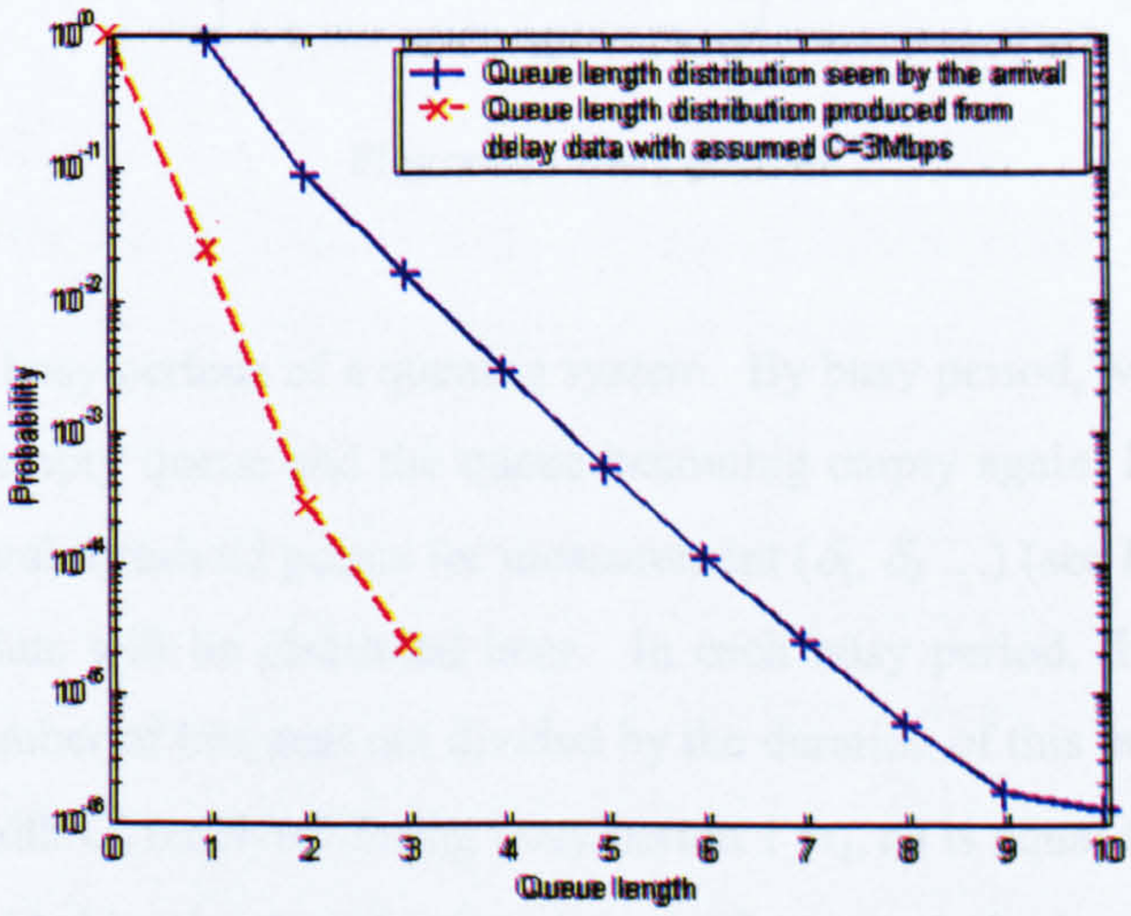


Figure 6-7 Comparison between the queue length distribution and the delay

If we assume the received bandwidth is 3M bps for queue#1. With Equation 4-26, the packet delay can be represented in terms of queue size. This is indicated by the red line in Figure 6-7. Compared with the actual queue length distribution seen by the incoming packets in queue#1, it can be noted that there is a discrepancy between the two distributions. This illustrates that the theoretical bandwidth $\phi_i C$ is actually smaller than the actual received bandwidth and leads to inaccurate representation of delay distribution from the queue length distribution. To solve this problem, we introduce a measurement mechanism called BW Measurement for bandwidth estimation.

6.3 Measurement for Bandwidth Estimation (BW Measurement)

In this section, we propose a passive/non-intrusive measurement scheme to estimate the actual bandwidth received in a sub-queue of a work conserving scheduler. Similar to QL Measurement, only addition and comparison operations are present in BW Measurement scheme to maintain the simplicity of the measurement method.

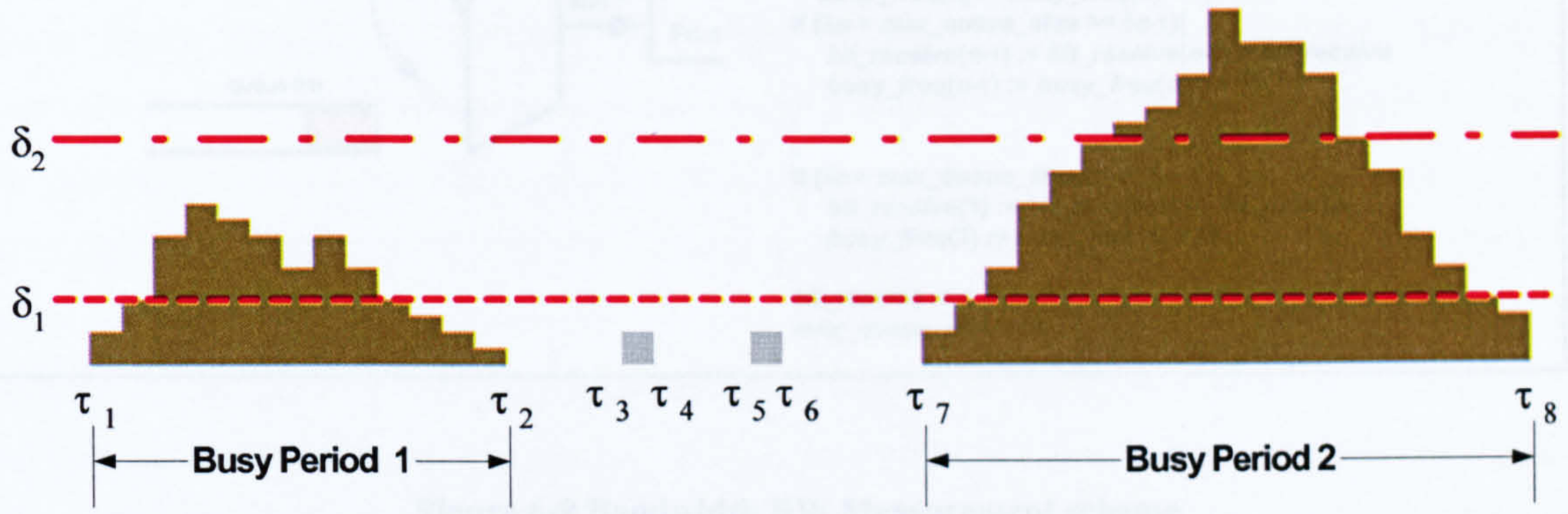


Figure 6-8 Busy periods

Figure 6-8 depicts the busy periods of a queuing system. By busy period, we mean the time between the first arrival to an empty queue and the queue becoming empty again. In our BW Measurement scheme, there are several threshold points for measurement (δ_1, δ_2 …) (see Figure 6-8). The purpose of these threshold points will be discussed later. In each busy period, the bandwidth received is simply equal to the number of bits sent out divided by the duration of this busy period. For example, the equivalent bandwidth C_e received during busy period 1 (τ_1, τ_2) is equal to number of bit received in this busy period divided by the time duration $(\tau_2 - \tau_1)$. In general, the longer the queue length gets, the longer the busy period will be.

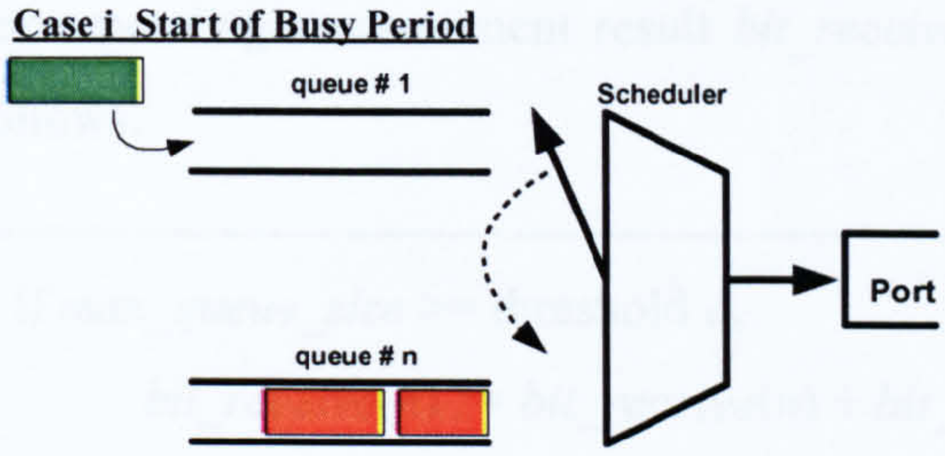


Figure 6-9 Bandwidth BW Measurement scheme

Figure 6-9 depicts the Bandwidth Measurement mechanism and its pseudo code. This measurement scheme is performed as follows. There are three measurement data: *bit_receive*(*n*), *busy_period*(*n*) and *busy_freq*(*n*) for each measuring threshold δ_n. The boolean variable *busy* serves as an indicator of the queue's status. When a busy period starts, i.e. at the moment an empty queue is seen by an arrival, the variable *busy_start_time* marks the present time when this event happens. During the busy period, the variables *max_queue_size* and *bit_receive* are updated as shown in the pseudo code. At the end of the busy period, i.e. the last packet in the queue has been served, the variable *bit_receive* stores the value of the number of bits served. The variable *max_queue_size* is the largest queue length encountered in this busy period. When the queue becomes empty again, the difference

between the current time and *busy_start_time* tells the duration of this busy period which is stored in variable *busy_period*. The variable *max_queue_size*, the largest queue size seen in this busy period, is compared to every measuring threshold δ_n. If *max_queue_size* is greater than the threshold δ_n, the corresponding measurement result *bit_receive*(*n*), *busyeq*(*n*), *busy_duration*(*n*) will be updated as follows.

```
if max_queue_size >= threshold δ_n
    bit_receive(n) := bit_receive(n) + bit_receive
    busy_duration(n) := busy_duration(n) + busy_period
    busy_freq(n) := busy_freq(n) + 1
```

For example, in Figure 6-8, the maximum queue length during busy period 1 and 2 exceeded the threshold δ_1 and δ_2, but the queue length in busy period 1 is always below the threshold δ_2. Therefore, the data taken in both busy period 1 and 2 are used to update the measurement results for threshold δ_1 and δ_2, but the data taken in busy period 1 will be excluded from the measurement result for δ_2. The variable *busy_freq*(*n*) is the number of occurrences of the busy periods for which of the queue length exceeds the threshold δ_n. This variable serves as an index for selecting the measurement result. The longer the queue length during the busy period is, the longer duration of this period is likely to be. Therefore, the measurement result corresponding to a larger threshold is more representative for determining the actual received bandwidth. However, the trade-off is that the occurrence of the busy periods with longer queue lengths is rare, and so is the measurement data. As a result, the variable *busy_freq*(*n*) is used to determine which measurement data *bit_receive*(*n*) and *busy_duration*(*n*) should be chosen. In our measurement scheme, we select the measurement data with the highest threshold *n* provided that its *busy_freq*(*n*) exceeds a criterion[18], say 100. Then the received bandwidth can be estimated as:

$$Ce = \frac{bit\_receive(n)}{busy\_time(n)} \qquad \max n : busy\_freq(n) \geq 100 \qquad \textbf{Equation 6-3}$$

[18] The reason behind using 100 as the selection criterion is that the tail end of the queue state distribution is noisy due to rare events happened in that region [STE02]. By using 100 as the selection criterion, the observation of the queue length away from the end of the tail will be included in the measurement data. If a small selection criterion is used, then only few busy periods' measurement data is used for bandwidth estimation which may not be representative for the whole bandwidth received in the period. However, if a large selection criterion is used, those busy periods with small queue length is included. When the queue length is small, it is most likely for an arriving packet sees an empty queue and is served immediately. Then, the estimated bandwidth will be biased on the link bandwidth.

6.4 Simulation Model – Single Hop

We performed simulations to validate our BW Measurement methodology. For brevity, we assume that there are only two traffic classes. For multiple traffic classes, if we consider the spare bandwidth of the multiple traffic classes used by a particular traffic class of interest, these multiple traffic classes can be reduced into a single traffic class as shown in Figure 6-10. Therefore, it is fair enough to assume that there exist two traffic classes in our experiment.

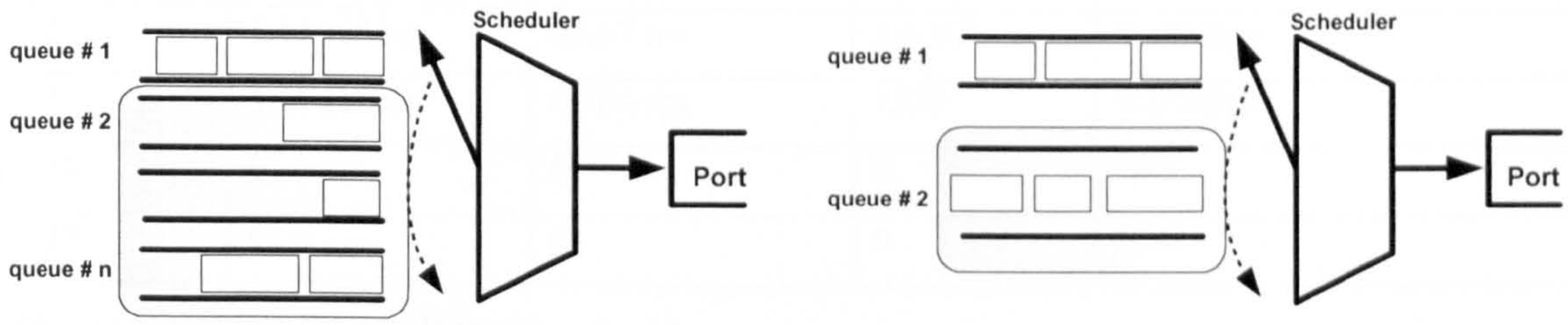


Figure 6-10 Multiple and two traffic classes

In the following experiments, two traffic classes are fed with multiple ON/OFF sources. It is assumed queue #1 is of interest and we perform our measurement methodology to estimate the received bandwidth.

6.4.1 Simulation Results (BW Measurement)

6.4.1.1 *Scenario: WRR Scheduling - two Markovian traffic classes*

In this experiment, the set-up is the same as that in Table 6-1(B). WRR is employed as the scheduler. The measuring thresholds δ_n are 0, 2, 5, 10 and 20. We perform the measurement for 3600 sec.

Traffic class	#1	#2
On Time	0.96sec (exponential)	0.96sec (exponential)
OFF Time	1.69sec (exponential)	1.69sec (exponential)
Number of sources	100	30
bit rate	70808bps	708080bps
weight	1	5
service rate	18Mbps	

Table 6-2 Experimental set-up for scenario 6.1

We assume that one of the 100 ON/OFF sources in traffic class #1 is of interest. This ON/OFF source is still referred to as foreground traffic. We collect the packet delay data of this foreground traffic. The packet delays are converted into the queue length distribution with the theoretical bandwidth $\phi_1 C$ and the estimated bandwidth respectively. These are compared with the queue length distribution actually seen by arrivals. Table 6-3 shows the measurement result.

threshold (n)	*bit_receive*(n)	*busy_period*(n)	*busy_freq*(n)	*bit_receive*(n)/*busy_period*(n)
$\delta_1 = 0$	$9.25 \cdot 10^9$ bit	822.65 sec	$1.69 \cdot 10^7$	11.24Mbps
$\delta_2 = 2$	$5.14 \cdot 10^8$ bit	65.67 sec	$2.6 \cdot 10^7$	7.82Mbps
$\delta_3 = 5$	$5.46 \cdot 10^6$ bit	0.794 sec	1320	6.88Mbps
$\delta_4 = 10$	0	0	0	N/A
$\delta_5 = 25$	0	0	0	N/A

Table 6-3 Measurement result for Senario 6.1

Table 6-3 shows the measurement result with different threshold values. As we discussed in Section 6.3, the measurement result of threshold δ_3 is chosen which gives the estimated bandwidth = 6.88Mbps.

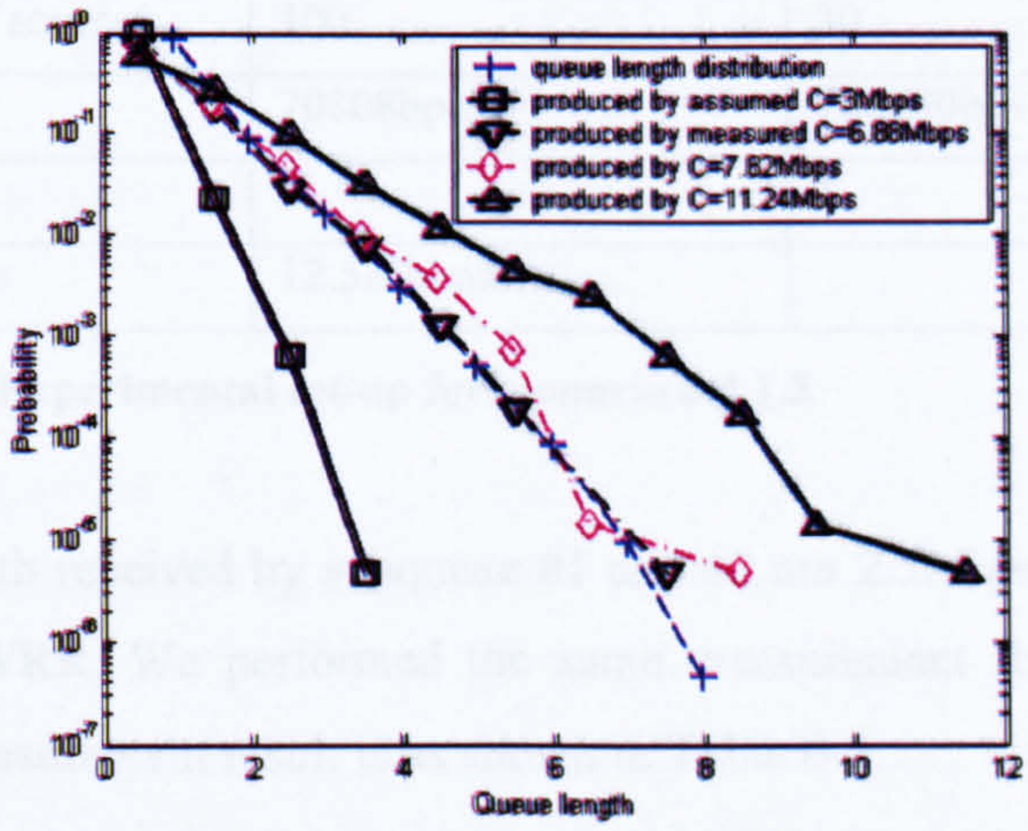


Figure 6-11 Comparison between the queue length distribution and the delay

Figure 6-11 shows the queue length distributions. With experimental set-up as shown in Table 6-2, the theoretical bandwidth allocated to traffic class #1 is supposed to be one-sixth of 18Mbps, i.e. 3Mbps that results in 0.855 utilization. Under this condition, a burst-scale queuing effect would be expected. On the contrary, only packet-scale queuing appears in Figure 6-11 and so the subqueue #1 should not be heavy-loaded as anticipated. The actual received bandwidth should be much greater than the theoretical bandwidth $\phi_1 C$ =3Mbps. We converted the packet delay statistic to

the queue length distribution by using Equation 4-26 with respect to the theoretical bandwidth (3Mbps) and the measured bandwidth (6.88Mbps). With reference to Figure 6-11, it can be noted that the black line, using $\phi_1 C$, has a bigger discrepancy with the queue length distribution seen by arrivals. However, the red line based on the measured bandwidth produces a better result. Since the received bandwidth is approximately double compared with the assumed bandwidth, the load actually is $100 \cdot 70808 \cdot 0.96 / \{(0.96+1.69) \cdot 6.88M\} = 0.37$. This low utilization condition accounts for only the presence of packet-scale queuing.

6.4.1.2 *Scenario: WRR Scheduling - Markovian traffic with 2 classes*

In this experiment, the utilization of subqueues #2 is deliberately set to high such that small amount of stolen bandwidth by traffic class #1 is expected. Table 6-4 shows the experimental set-up.

Traffic class	#1	#2
On Time	0.96sec (exponential)	0.96sec (exponential)
OFF Time	1.69sec (exponential)	1.69sec (exponential)
Number of sources	100	30
bit rate	70808bps	708080bps
weight	2	8
service rate	12.5Mbbps	

Table 6-4 Experimental set-up for scenario 6.4.1.2

The theoretical bandwidth received by subqueue #1 and #2 are 2.5Mbps and 10Mbps respectively. The scheduler is still WRR. We performed the same measurement for bandwidth estimation of traffic class #1. The measurement result is as shown in Table 6-5.

threshold (n)	*bit_receive*(n)	*busy_period*(n)	*busy_freq*(n)	*bit_receive*(n)/*busy_period*(n)
$\delta_1 = 0$	$9.24 \cdot 10^9$ bit	1930.11 sec	$1.02 \cdot 10^7$	4.78Mbps
$\delta_2 = 2$	$3.626 \cdot 10^9$ bit	1093 sec	$7.1 \cdot 10^5$	3.3Mbps
$\delta_3 = 5$	$2.1 \cdot 10^9$ bit	723 sec	$4.6 \cdot 10^4$	2.9Mbps
$\delta_4 = 10$	$1.63 \cdot 10^9$ bit	594 sec	1538	2.7Mbps
$\delta_5 = 25$	$1.55 \cdot 10^9$ bit	564 sec	948	2.7Mbps
$\delta_6 = 50$	$1.46 \cdot 10^9$ bit	533 sec	280	2.7Mpbs

Table 6-5 Measurement result for Scenario 6.4.1.2

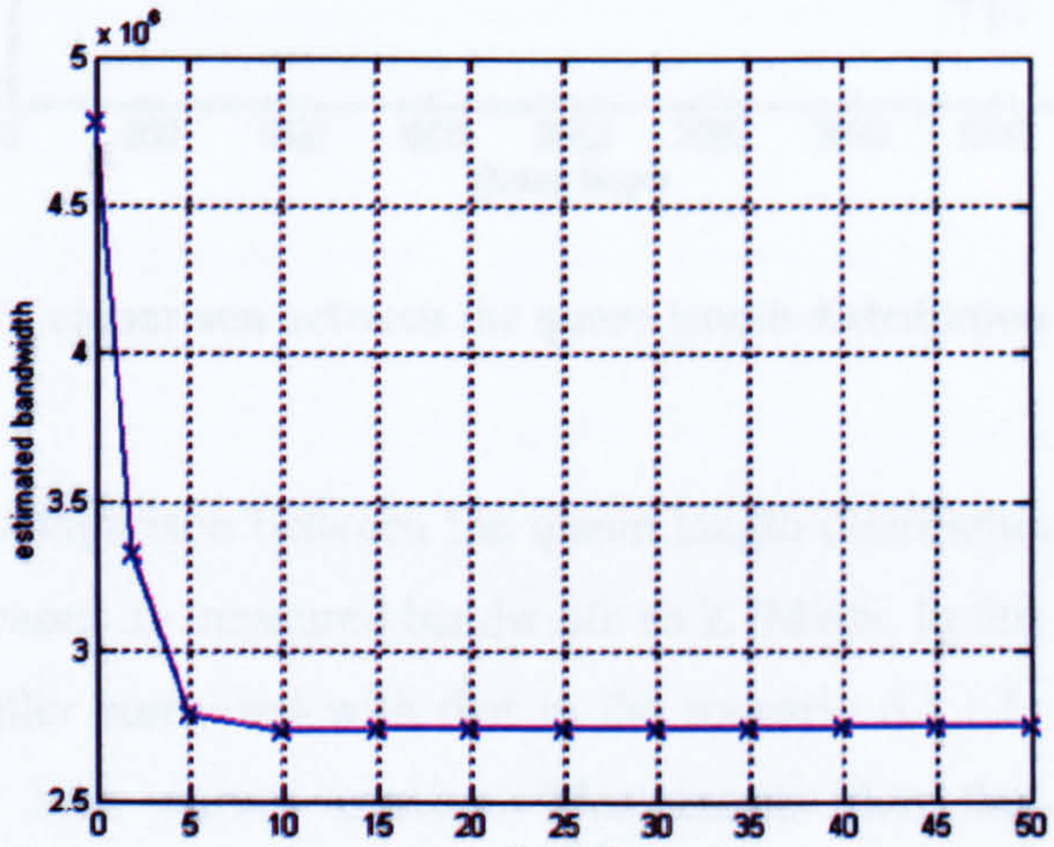


Figure 6-12 Estimated bandwidth versus threshold

Figure 6-12 shows the plot of estimated bandwidth versus different threshold values. The curve drops rapidly from the threshold value 0 to threshold value 5 and finally settles down to the value around 2.7Mbps. This demonstrates that the result is insensitive to the threshold point provided that a large threshold point is chosen. But the larger the threshold point is, the less chance the queue length exceeds this threshold, and so the less reliable the measurement result is. As a consequence, we heuristically set the threshold criterion as 100 (see Equation 6-3). Then, the estimated bandwidth in Table 6-5 will be given by δ_6 i.e. 2.7Mbps.

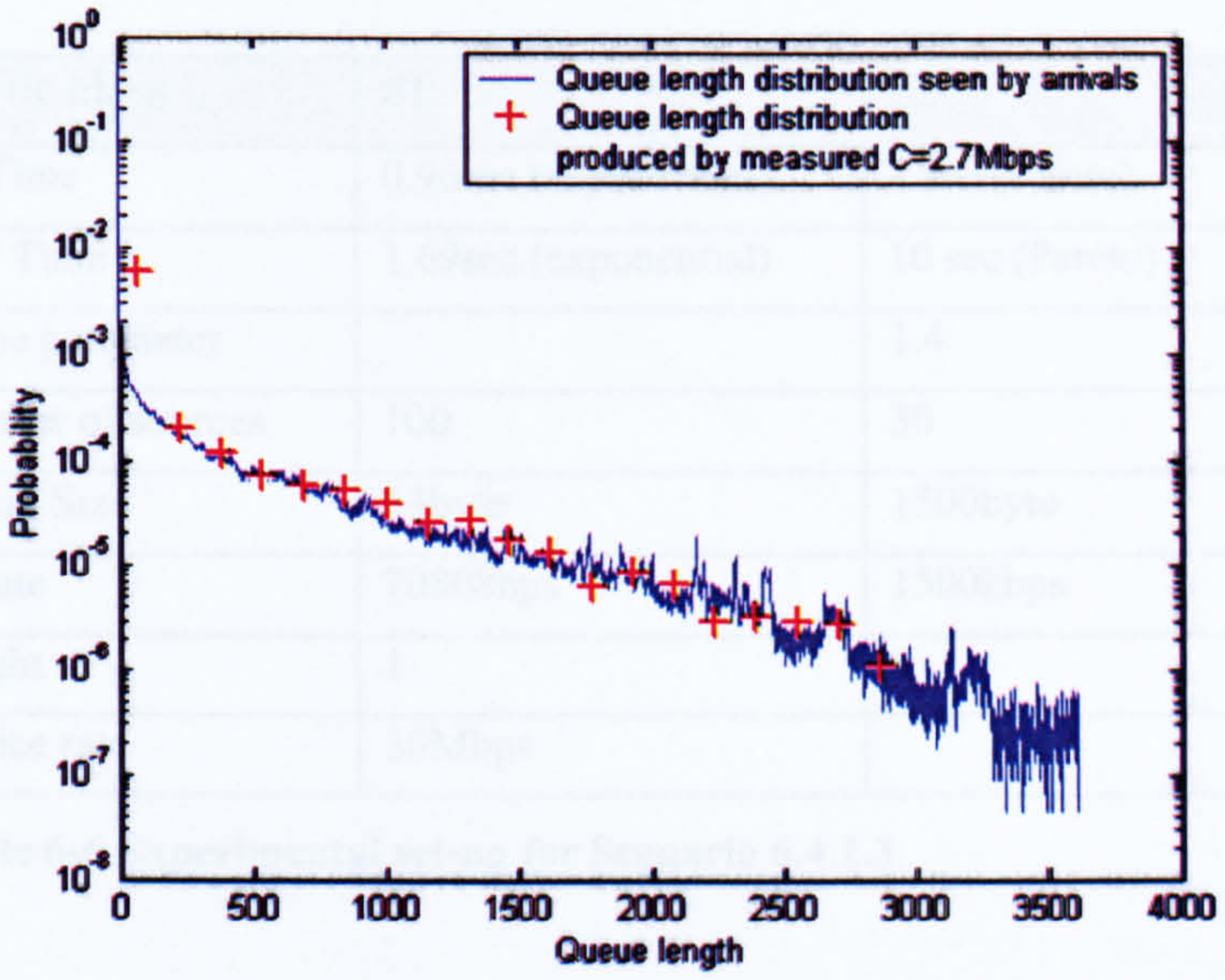


Figure 6-13 Comparison between the queue length distribution and the delay

Figure 6-13 shows the comparison between the queue length distribution seen by arrivals and that from the packet delays based on measured bandwidth as 2.7Mbps. In this case, the amount of stolen bandwidth is much smaller compared with that in the scenario 6.4.1.1. With reference to Figure 6-13, this shows that both curves overlap. This means that the Bandwidth Measurement methodology works quite well. Furthermore, it is worth pointing out that the tail of the queue length distribution still exhibits exponential decays. It means that the introduction of the scheduler does not alter the fundamental queuing characteristic. As a consequence, our QL Measurement scheme is still applicable.

6.4.1.3 *Scenario: WFQ Scheduling - Markovian traffic and Power-law traffic*

In this experiment, the second traffic class is driven by the Power-law traffic which is more bursty than Markovian traffic. The objective of this experiment is to study the effectiveness of the methodology under very bursty traffic condition [19]. The experimental set-up is shown as Table 6-6.

---

[19] In [GAR01], it highlights that the wrong aggregation of traffic flows with different statistical features, may lead to performance worsening, which should be avoided. As a consequence, in this thesis, the two different traffic, Markovian and Power-law traffic, are assumed to be separated by the scheduler.

Traffic class	#1	#2
On Time	0.96sec (exponential)	1 sec (Pareto)
OFF Time	1.69sec (exponential)	10 sec (Pareto)
Shape parameter		1.4
Number of sources	100	30
Packet Size	53byte	1500byte
bit rate	70808bps	1500kbps
weight	1	9
service rate	30Mbps	

Table 6-6 Experimental set-up for Scenario 6.4.1.3

WFQ is employed in this experiment as it is more effective than WRR in the regime of variable packet size.

threshold (n)	*bit_receive*(n)	*busy_period*(n)	*busy_freq*(n)	*bit_receive*(n)/*busy_period*(n)
$\delta_1 = 0$	$9.25 \cdot 10^9$ bit	626.54 sec	$1.8 \cdot 10^7$	14.76Mbps
$\delta_2 = 2$	$9.9 \cdot 10^8$ bit	210.9 sec	$5.4 \cdot 10^5$	4.69Mbps
$\delta_3 = 5$	$1.46 \cdot 10^8$ bit	24 sec	45838	6.08Mbps
$\delta_4 = 10$	48760	0.005 sec	9	9.752Mbps
$\delta_5 = 25$	0	0	0	N/A

Table 6-7 Measurement result for Scenario 6.4.1.3

According to the measurement result, the estimated bandwidth received by traffic class #1 is 6.08Mbps. A significant amount of bandwidth is stolen from traffic class #2.

Figure 6-14 shows the comparison. Similar to the scenario 6.4.1.1, the burst-scale queuing behaviour does not exist due to extra bandwidth gained. Again, the queue length distribution produced by the packet delay based on the measured bandwidth (red line) closely matches with that seen by arrivals. However, the queue length distribution produced by the packet delay based on theoretical bandwidth (black line), i.e. 3Mbps, gives poorer estimation result for packet class #1.

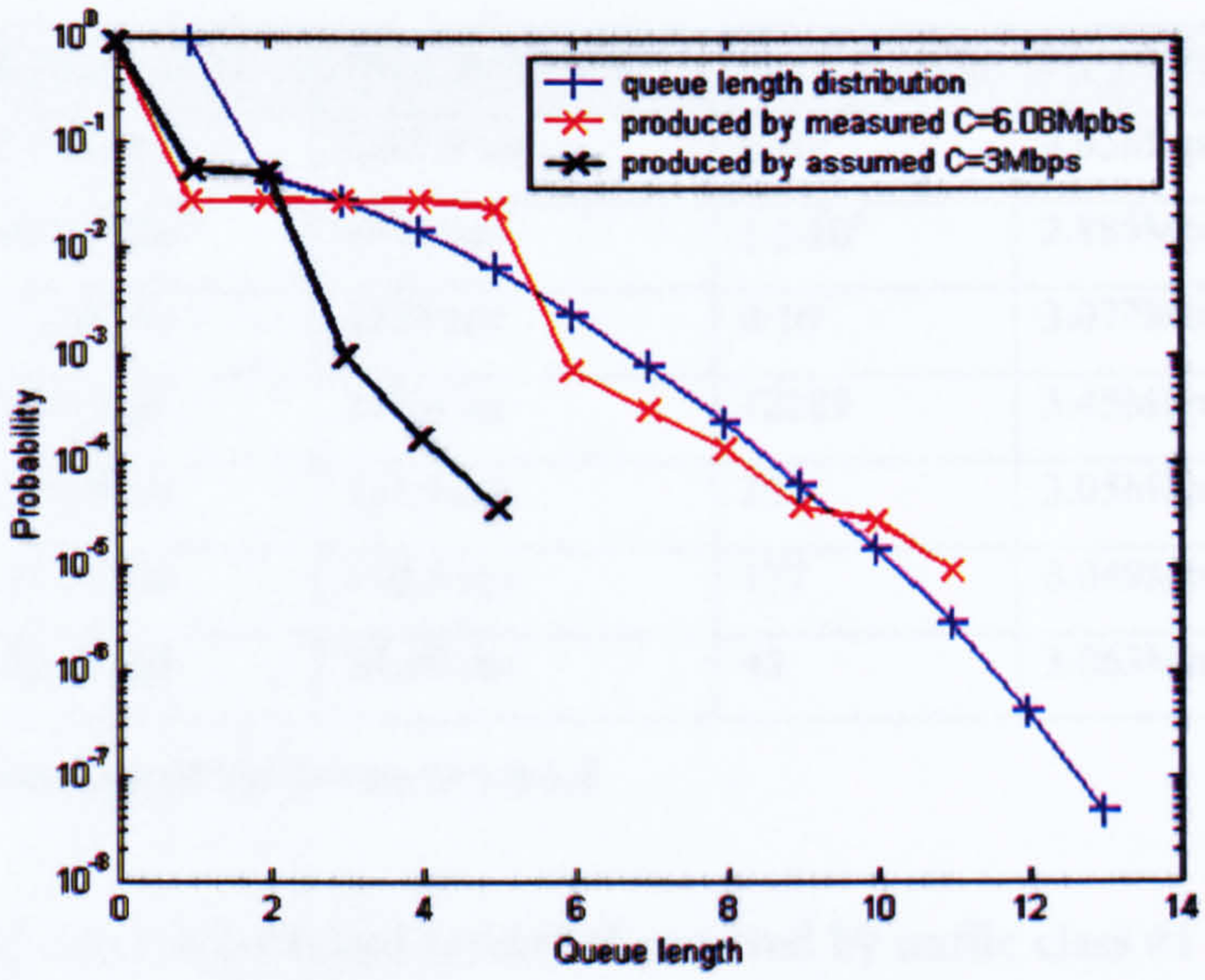


Figure 6-14 Comparison between the queue length distribution and the delay

6.4.1.4 *Scenario: WFQ Scheduling - Markovian traffic and Power-law traffic*

Similar to the scenario 6.4.1.2, we deliberately increase the load in subqueue #2 but the subqueue #2 is driven by Power-law traffic instead. The traffic parameters are same as in scenario 6.4.1.3. WFQ is still employed in this experiment.

Traffic class	#1	#2
On Time	0.96sec (exponential)	1 sec (Pareto)
OFF Time	1.69sec (exponential)	10 sec (Pareto)
Shape parameter		1.4
Number of sources	100	30
Packet Size	53byte	1500byte
bit rate	70808bps	1500kbps
weight	3	7
service rate	10Mbbps	

Table 6-8 Experimental set-up for Scenario 6.4.1.4

threshold (n)	*bit_receive*(n)	*busy_period*(n)	*busy_freq*(n)	*bit_receive*(n)/*busy_period*(n)
$\delta 1 = 0$	$9.25 \cdot 10^9$ bit	2283.9 sec	$8 \cdot 10^6$	4.05Mbps
$\delta 2 = 2$	$5.43 \cdot 10^9$ bit	1882 sec	$1.2 \cdot 10^6$	2.885Mbps
$\delta 3 = 5$	$4.65 \cdot 10^9$ bit	1524 sec	$9 \cdot 10^5$	3.077Mbps
$\delta 4 = 10$	$9.24 \cdot 10^8$ bit	267.6 sec	72289	3.45Mbps
$\delta 5 = 25$	$3.81 \cdot 10^8$ bit	124.9 sec	252	3.05Mbps
$\delta 6 = 50$	$3.37 \cdot 10^8$ bit	110.5 sec	177	3.049Mbps
$\delta 6 = 250$	$1.66 \cdot 10^8$ bit	54.19 sec	43	3.063Mbps

Table 6-9 Measurement result for Scenario 6.4.1.4

According to Table 6-9, the estimated bandwidth received by traffic class #1 is 3.049Mbps. Under this scenario, there is only a small amount of bandwidth stolen from traffic class #2. Figure 6-15 shows the results of queue length distributions for packet class #1.

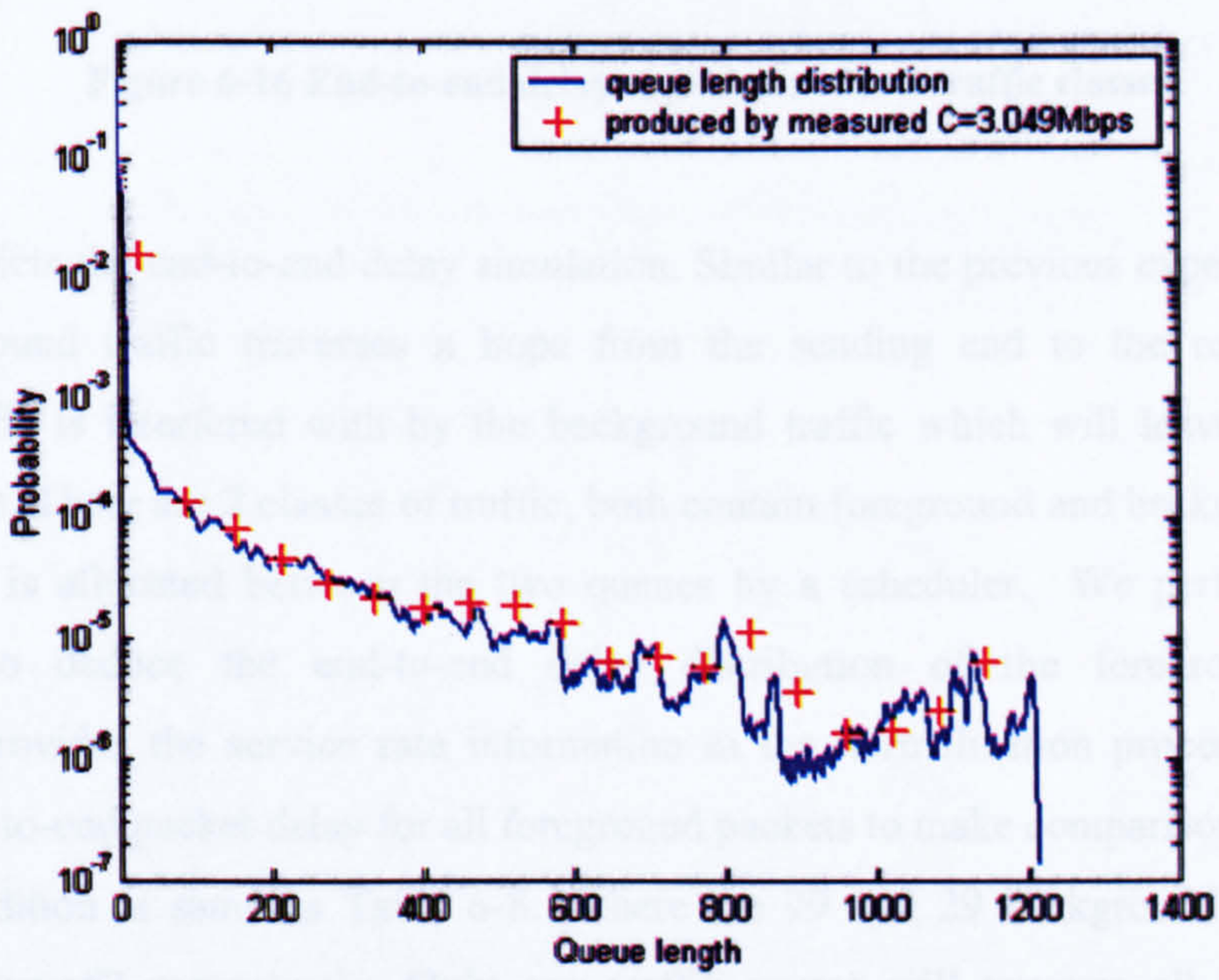


Figure 6-15 Comparison between the queue length distribution and the delay

With reference to Figure 6-15, it can be seen that a burst-scale queuing effect exists and produces a relatively longer tail in queue length distribution compared with that in Figure 6-14. Again, the queue length distribution produced from the packet delay based on the measured bandwidth and that seen by arrivals closely match with each other.

6.4.2 Simulation Results (End-to-end Delay Measurement)

The evaluation of BW Measurement scheme was presented in the previous section. In this section, BW Measurement scheme is integrated into our QL Measurement scheme for the end-to-end delay distribution measurement.

6.4.2.1 *Scenario: End-to-end simulation 4hops and 10hops*

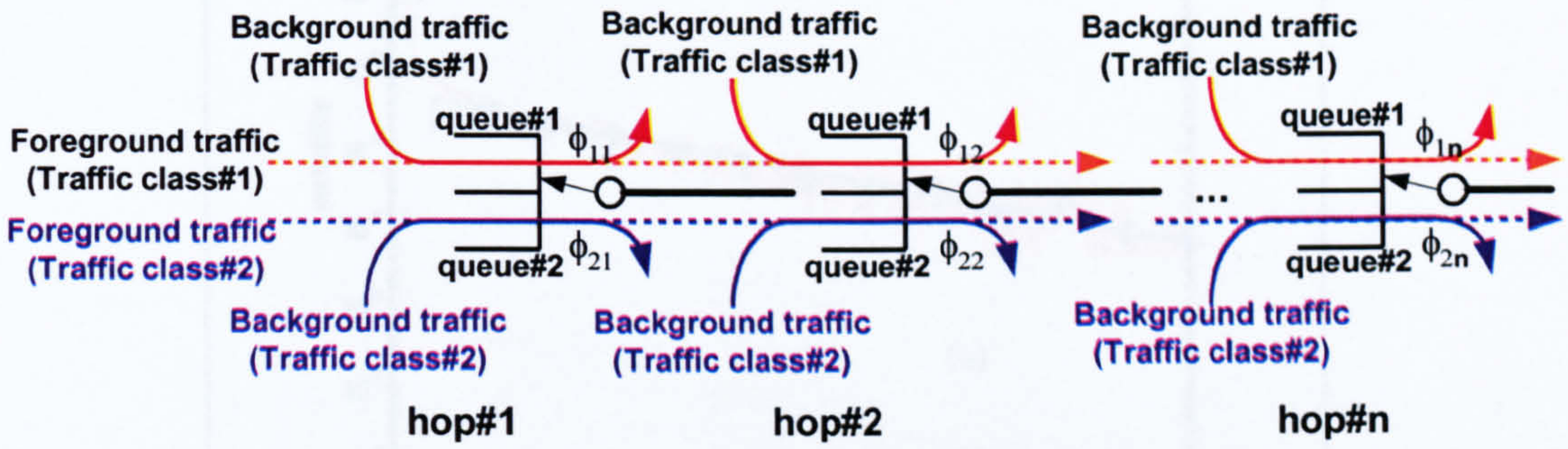


Figure 6-16 End-to-end delay experiment for 2 traffic classes

Figure 6-16 depicts the end-to-end delay simulation. Similar to the previous experiments in Section 5.5, the foreground traffic traverses n hops from the sending end to the receiving end. The foreground traffic is interfered with by the background traffic which will leave the system after passing the node. There are 2 classes of traffic, both contain foreground and background traffic. The link bandwidth is allocated between the two queues by a scheduler. We perform QL and BW Measurement to deduce the end-to-end delay distribution of the foreground traffic. BW Measurement provides the service rate information in the normalization process. We collect the statistics of end-to-end packet delay for all foreground packets to make comparison with the estimate. The traffic condition is same as Table 6-8. There are 99 and 29 background traffic sources in queue#1 and queue#2 respectively. Only one traffic source will traverse all hops to reach the receiving end. The link bandwidths are 10Mbps and the weights are 3 and 7 for queue#1 and queue#2 respectively. The simulation results for 4 hops and 10 hops are presented. The measured bandwidth for a 4-hop case is shown in Table 6-10.

	hop#1	hop#2	hop#3	hop#4
queue#1	3.067Mbps	3.075Mbps	3.053Mbps	3.045Mbps
queue#2	7.4Mbps	7.367Mbps	7.385Mbps	7.39Mbps

Table 6-10 Bandwidth measurement result (4 hops)

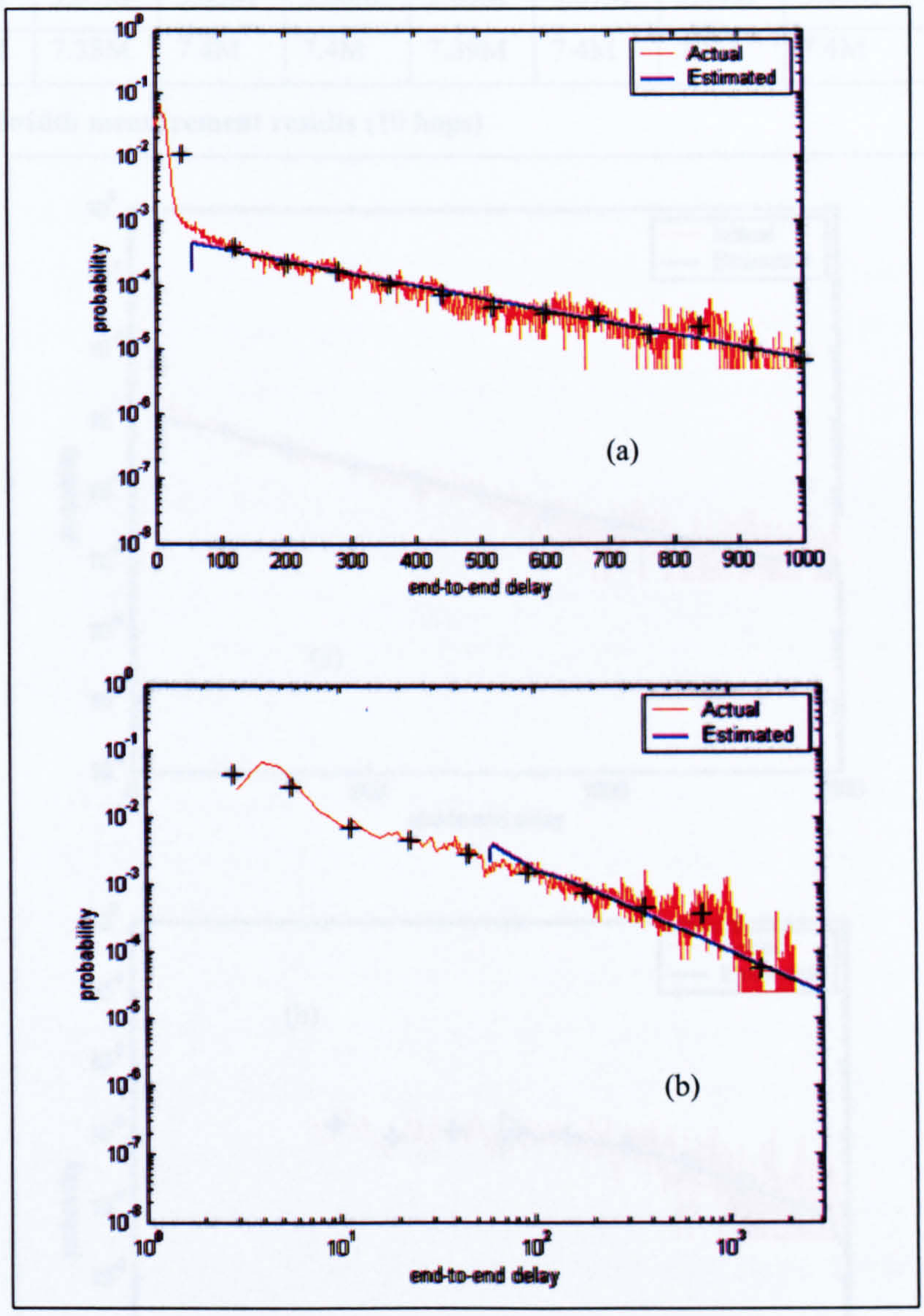


Figure 6-17 End-to-end delay distribution (4 hops case) (a) Foreground Traffic#1 (b) Foreground Traffic#2

Figure 6-17 shows the comparison of results between the true and estimated end-to-end delay distribution. It can be seen that the tail of the end-to-end delay distribution are accurately estimated. We repeat the experiment with the same traffic conditions for 10 hops. The measured bandwidth for a 10 hops case is shown in Table 6-11.

	hop#1	hop#2	hop#3	hop#4	hop#5	hop#6	hop#7	hop#8	hop#9	hop#10
queue#1	3.07M	3.07M	3.03M	3.06M	3.05M	3.07M	3.04M	3.05M	3.07M	3.04M
queue#2	7.39M	7.38M	7.4M	7.4M	7.39M	7.4M	7.4M	7.4M	7.38M	7.4M

Table 6-11 Bandwidth measurement results (10 hops)

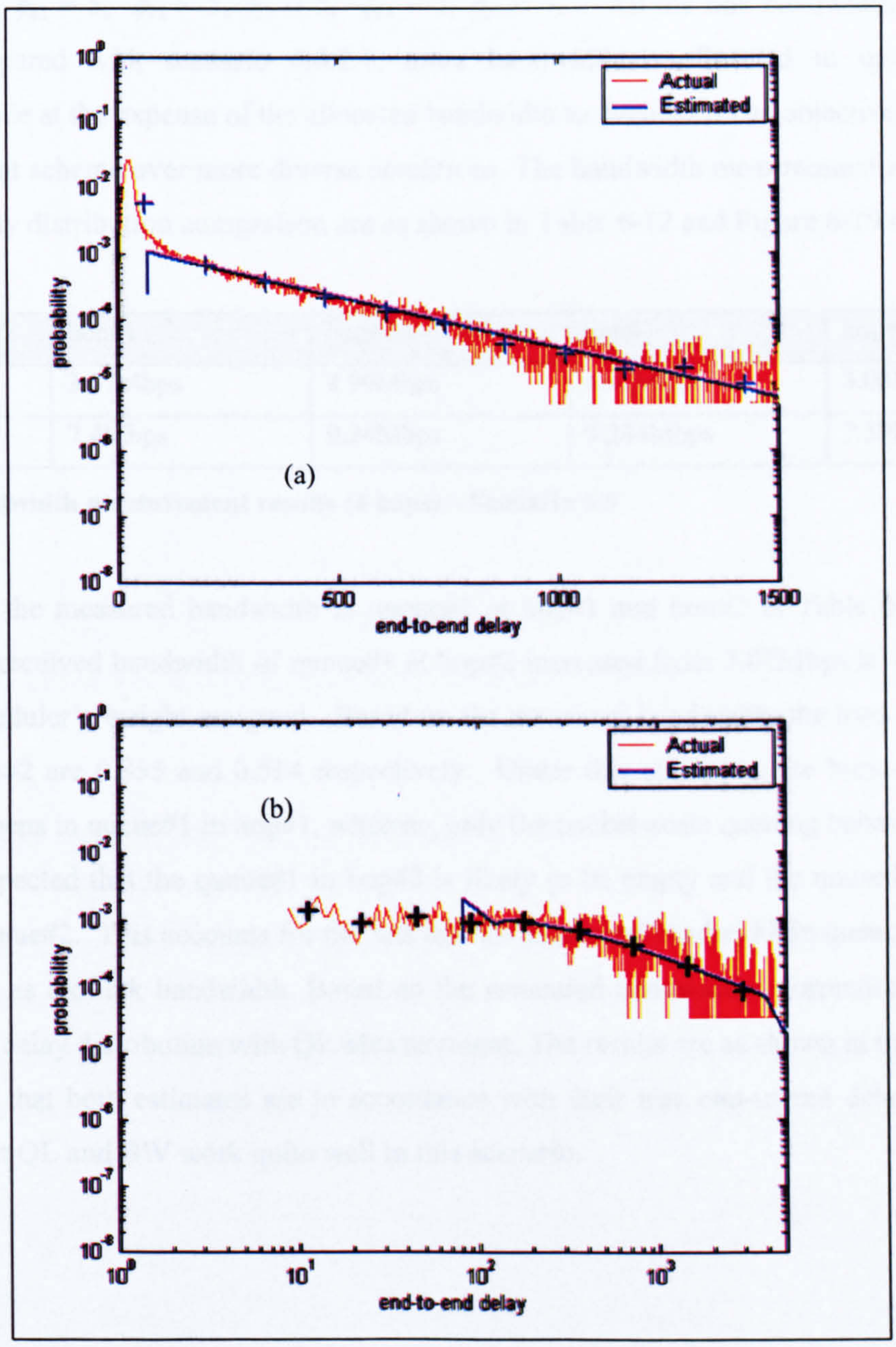


Figure 6-18 End-to-end delay distribution (10 hops case) (a) Foreground Traffic#1 (b) Foreground Traffic#2

Figure 6-18 shows the results of end-to-end delay measurement for an end-to-end path that is 10 hops long. It can be seen that the measurement method gives the estimation result with accuracy.

6.4.2.2 *Scenario: 4-hops (various scheduler weights assignment)*

In this experiment, the foreground traffic traverses 4 hops before reaching the receiving end. Various scheduler weights are assigned in the following way: ϕ_{11} (the weight of queue#1 in hop#1) = 3, ϕ_{21}=7, ϕ_{12} = 5, ϕ_{22} = 5, ϕ_{13} = 5, ϕ_{23} = 5, ϕ_{14} = 3, ϕ_{24} = 7. All the link bandwidths are equal to 10Mbps. Compared with scenario 6.4.2.1, more bandwidth is allocated to queue#1 in the intermediate node at the expense of the allocated bandwidth to queue#2. The objective is to evaluate the measurement scheme over more diverse conditions. The bandwidth measurement results and the end-to-end delay distribution comparison are as shown in Table 6-12 and Figure 6-19 respectively.

	hop#1	hop#2	hop#3	hop#4
queue#1	3.07Mbps	4.99Mbps	4.92Mbps	3.08Mbps
queue#2	7.4Mbps	9.24Mbps	9.244Mbps	7.39Mbps

Table 6-12 Bandwidth measurement results (4 hops) – Scenario 6.5

By comparing the measured bandwidth in queue#1 at hop#1 and hop#2 in Table 6-12, it can be found that the received bandwidth of queue#1 at hop#2 increased from 3.07Mbps to 4.99Mbps due to a larger scheduler's weight assigned. Based on the measured bandwidth, the load of queue#1 at hop#1 and hop#2 are 0.855 and 0.514 respectively. Under this condition, the burst-scale queuing behaviour happens in queue#1 in hop#1, whereas, only the packet-scale queuing behaviour occurs in hop#2. It is expected that the queue#1 in hop#2 is likely to be empty and the unused bandwidth is given to the queue#2. This accounts for the fact that the measured bandwidth in queue#2 at hop#2 is almost as high as the link bandwidth. Based on the measured bandwidth information, we estimate the end-to-end delay distribution with QL Measurement. The results are as shown in the Figure 6-19. It can be seen that both estimates are in accordance with their true end-to-end delay distribution. This shows that QL and BW work quite well in this scenario.

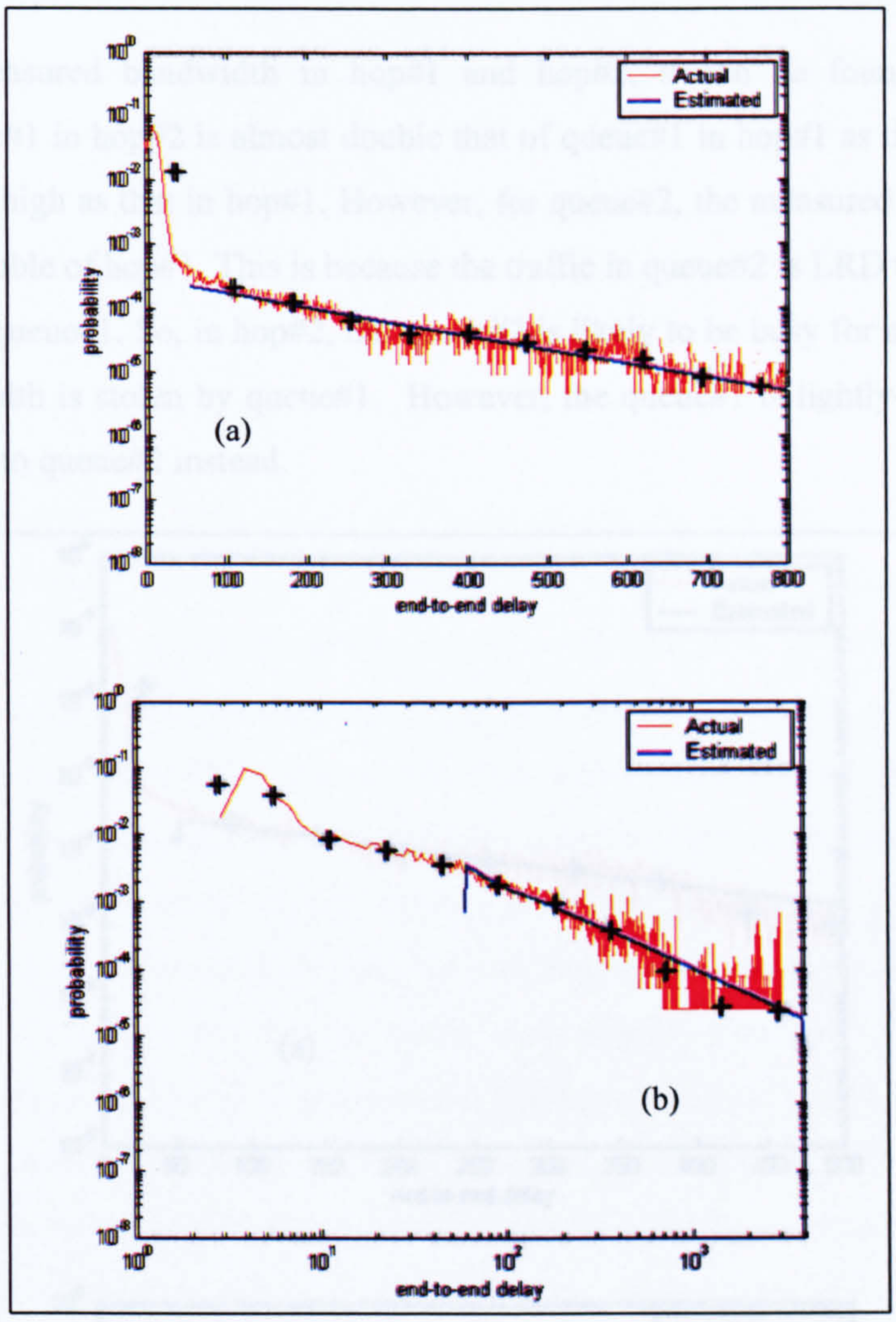


Figure 6-19 End-to-end delay distribution (a) Foreground Traffic#1 (b) Foreground Traffic#2

6.4.2.3 *Scenario: 4-hops with different link bandwidths*

In this experiment, there are 4 hops in the end-to-end path. The link bandwidths are various along the path: C_1 = 10Mbps, C_2 = 20Mbps, C_3 = 20Mbps, C_4 = 10Mbps. The scheduler weights are set as queue#1 is 3 whereas, queue#2 is 7. The bandwidth measurement results and the end-to-end delay distribution comparison are as shown in Table 6-13 and Figure 6-20 respectively.

	hop#1	hop#2	hop#3	hop#4
queue#1	3.51Mbps	6.9Mbps	6.9Mbps	3.06Mbps
queue#2	7.4Mbps	17.8Mbps	17.76Mbps	7.38Mbps

Table 6-13 Bandwidth measurement results (4 hops) – Scenario 6.4.2.3

Comparing the measured bandwidth in hop#1 and hop#2, it can be found that the received bandwidth of queue#1 in hop#2 is almost double that of queue#1 in hop#1 as the link bandwidth in hop#2 is double as high as that in hop#1. However, for queue#2, the measured bandwidth in hop#2 is more than the double of hop#1. This is because the traffic in queue#2 is LRD traffic which is more bursty than that in queue#1. So, in hop#2, the queue#2 is likely to be busy for most of the time. Not much extra bandwidth is stolen by queue#1. However, the queue#1 is lightly-loaded and so more bandwidth is given to queue#2 instead.

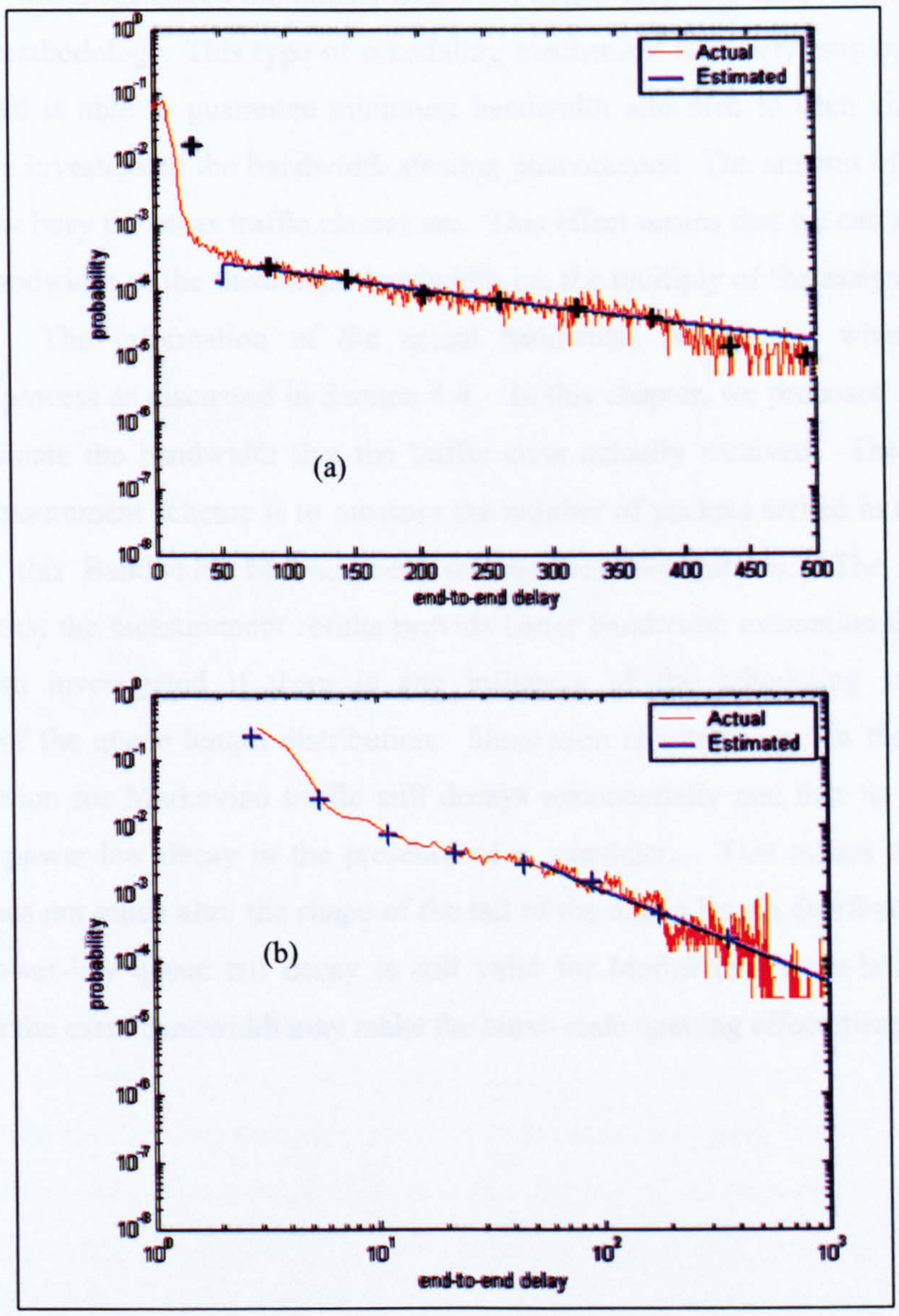


Figure 6-20 End-to-end delay distribution (a) Foreground Traffic#1 (b) Foreground Traffic#2

Figure 6-20 shows the result of end-to-end delay distribution by BW+QL Measurement for scenario 6.4.2.3. Again, it can be seen that BW+QL Measurement scheme successfully capture the end-to-end delay distribution in this scenario.

6.5 Summary

In this Chapter, we investigated the impact of a work conserving scheduler WRR and WFQ on our measurement methodology. This type of scheduling mechanism is widely employed for bandwidth partitioning, and is able to guarantee minimum bandwidth allocated to each class. Through the simulations, we investigated the bandwidth stealing phenomenon. The amount of stolen bandwidth depends on how busy the other traffic classes are. This effect means that we can not simply assume the received bandwidth as the theoretical bandwidth i.e. the multiply of the assigned weight and the link capacity. The information of the actual bandwidth is essential when performing the normalization process as discussed in Section 4.4. In this chapter, we proposed BW Measurement scheme to estimate the bandwidth that the traffic class actually received. The main idea of the Bandwidth Measurement scheme is to measure the number of packets served in every busy period. We evaluated this Bandwidth Measurement scheme with simulations. The simulation results demonstrated that the measurement results provide better bandwidth estimation than the theoretical one. We also investigated if there is any influence of the scheduling mechanism on the characteristic of the queue length distribution. Simulation results show that the tail of the queue length distribution for Markovian traffic still decays exponentially and that for Power-law traffic possesses the power-law decay in the presence of a scheduler. This means that the scheduling mechanism does not much alter the shape of the tail of the queue length distribution. Therefore, the exponential/power-law queue tail decay is still valid for Markovian/Power-law traffic. The only concern is that the extra bandwidth may make the burst-scale queuing effect disappear.

Chapter 7 Comparison with Active Probing

The current approach of end-to-end delay performance assessment in most practical networks is mainly based on active probing. We proposed a novel Passive Measurement scheme QL Measurement for end-to-end delay performance measurement that does not have any intrusive effect found in active probing as discussed in Chapter 4. In this section, we compare the end-to-end delay measurement performance of our scheme with active probing. In the experiments, a count-based active probing mechanism is employed to generate the testing packets. A testing packet is generated when pre-assigned amount of traffic is transmitted [ZSE02]. We compare the end-to-end delay distribution estimated by our scheme and active probing. In addition, we investigate the effectiveness of active probing and our measurement scheme on the SLA verification for the end-to-end delay performance.

7.1 Active Probing Mechanism

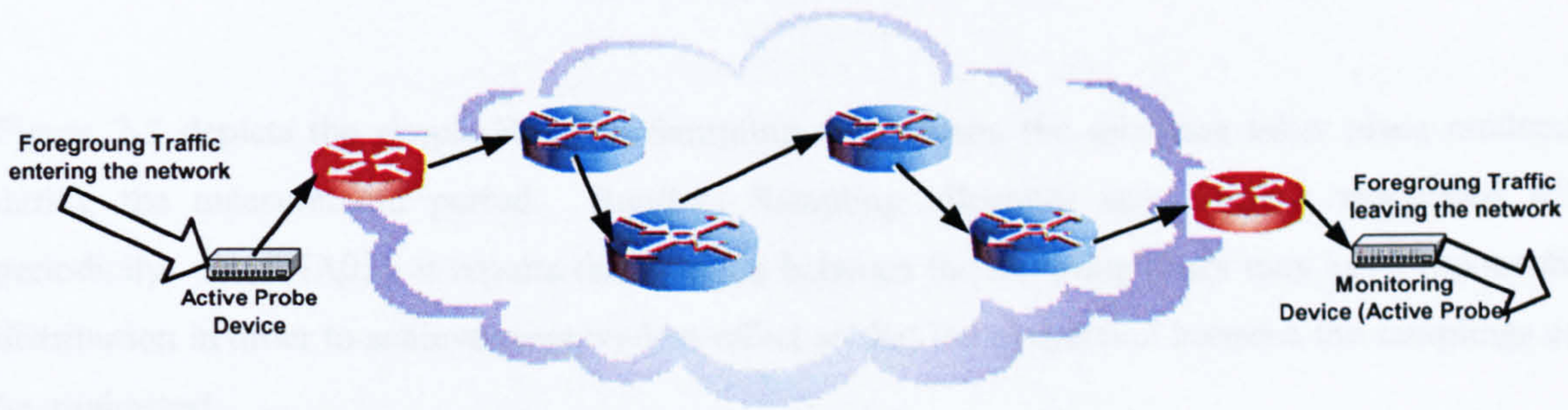


Figure 7-1 Active probing framework

Figure 7-1 depicts the active probing framework used in this thesis. The traffic of interest is referred to as the foreground traffic as in previous chapters. There exist active probing devices at both ends of the path which the foreground traffic traverses. At the sending side, the active probing device is responsible for generating the testing packets. These testing packets are time-stamped and received at the receiving end by a monitoring device. The monitoring device measures the testing packets' end-to-end delay. The statistic will be used to compare with its parent population statistics, i.e. foreground traffic.

Active probing is based on sampling. [ZSE02] addresses that "*the measurement costs usually should be limited to a small fraction of the costs of providing the network service itself, a reduction of the*

measurement result data is crucial to prevent the depletion of the available (i.e. the affordable) resources." Therefore, it is expected that the testing packet is a small fraction of the parent measured traffic[20].

7.1.1 Sampling Techniques

The purpose of the testing packet is to sample the end-to-end packet delay data from its parent population. In [HIL02], the author investigated various sampling techniques for QoS performance measurement, simple random, stratified random and system sampling [CLA93a]. These are described in more detail in the following sections.

7.1.2 Simple Random Sampling

Figure 7-2 Random Sampling

Figure 7-2 depicts the simple Random Sampling mechanism, the sampling takes place randomly during the measurement period. Random Sampling alleviates any problem associated with periodicity. In [SHA03], it reports the duration between the sampling times may have exponential distribution in order to achieve memory-less effect so that the correlation between the samplings can be minimized.

7.1.3 Stratified Random Sampling

Figure 7-3 Stratified Random Sampling

Stratified Random Sampling splits the population into *N* sub-populations. These populations do not overlap and they cover the whole of the original population. The sampling event takes place randomly within the sub-population. As for Simple Random Sampling, Stratified Sampling alleviates any problems associated with periodicity.

---

[20] In this thesis, the testing packet volume is not more than 1% of its parent traffic

7.1.4 Systematic Sampling

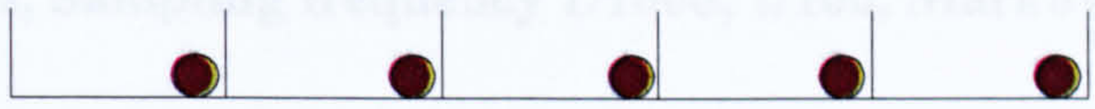


Figure 7-4 Systematic Sampling

Systematic Sampling takes place every *n* units as illustrated in Figure 7-4. This sampling method is employed in our experiments for the following reasons.

- Generally, the larger the sampling size, the greater the accuracy of the measurement result will be obtained. To study the performance of active probing, we would like to make the sampling size to be a fixed portion of the foreground traffic to limit the measurement cost as discussed above. Since the amount of the foreground traffic during the measurement period is usually unknown in advance, therefore, Systematic Sampling is easier to be implemented compared with Random Sampling if we want to limit the sampling size to be a fixed portion of the foreground traffic.

- Stratified Random Sampling is also able to ensure the probing traffic volume is a fixed portion of its parent traffic. Stratified Random Sampling provides better sampling result than Systematic Sampling provided that a priori knowledge of the correlation structure of the parent population is available [ZES02]. The splitting size as shown in Figure 7-4 becomes the size having small correlation in the parent population. Nevertheless, this priori-knowledge is not available during measurement.

With reference to Figure 7-1, the role of the active probing device at the sending end is to count the number of foreground traffic arrivals. With Systematic Sampling, a testing packet is generated by the active probing device for every N foreground traffic packets. If the testing packet size is comparable with the foreground traffic packet size, then the extra load caused by the active probing will be 1/N of the original foreground traffics as shown in Figure 7-5.

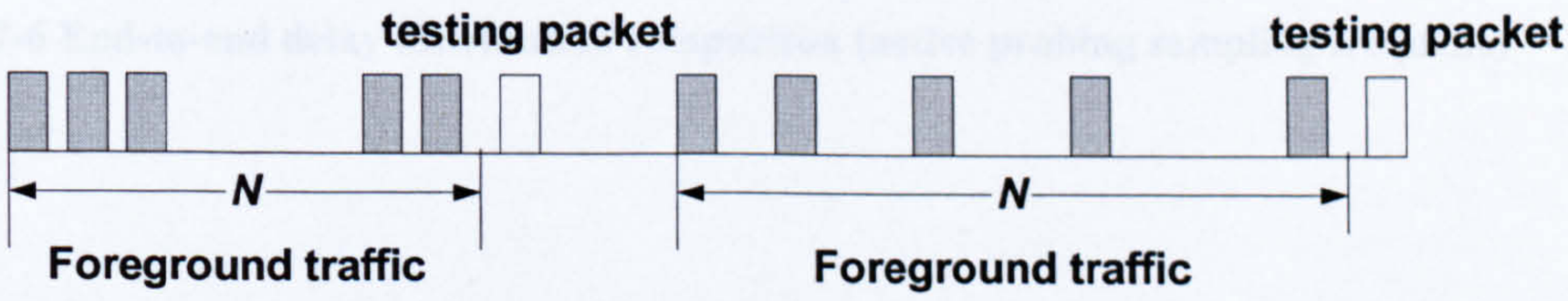


Figure 7-5 A testing packet is generated for every N foreground traffic

7.2 Simulation Results

7.2.1 Scenario: 4 hops, Sampling frequency 1/1000, 1/100, Markovian Traffic

The notation of the sampling frequency, $1/N$, means that the active probing device sends out one testing packet for every N foreground traffic packets. In this experiment, the foreground traffic traverses 4 hops before reaching the receiving end. The foreground traffic is monitored by the active probing device. This probing device sends out the testing packets with the sampling frequency = 1/1000 and 1/100 respectively. The experimental set-up is shown as follows.

Foreground traffic:

ON time: 0.96sec (Exponential), OFF time: 1.69sec (Exponential), Transmission rate: 167 pkt/sec, packet size: 53 byte

Background traffic:

Number of sources: 100, ON time: 0.96sec (Exponential), OFF time: 1.69sec (Exponential), Transmission rate: 167 pkt/sec, packet size: 53 byte

Service rate: S1: 3Mbps, S2: 3Mbps, S3: 3Mbps, S4: 3Mbps

Active Probing: Sampling frequency: 1/1000 foreground traffic, 1/100 foreground traffic

Measurement time: 3600 sec. Partition point: 50

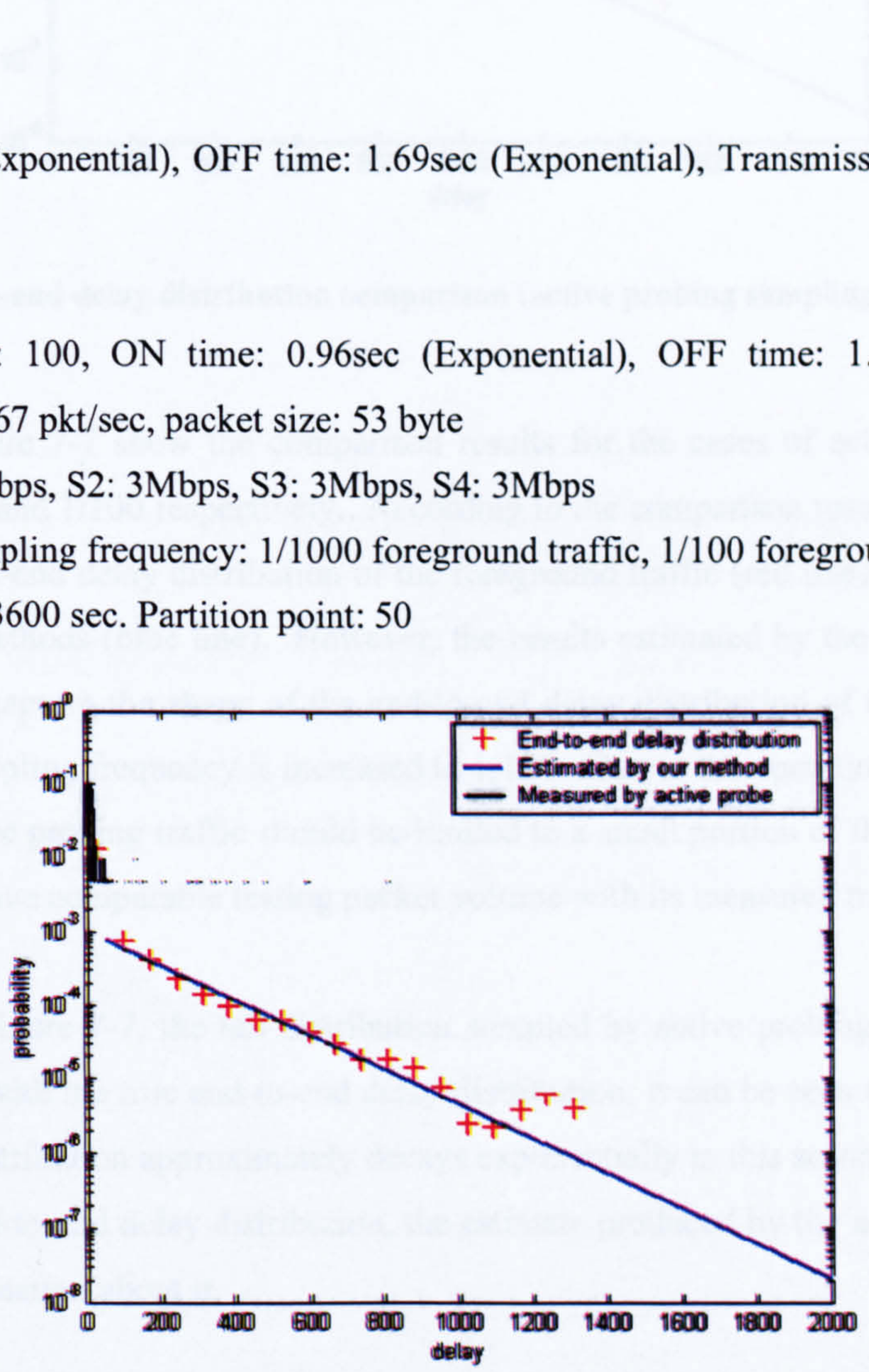


Figure 7-6 End-to-end delay distribution comparison (active probing sampling frequency = 1/1000)

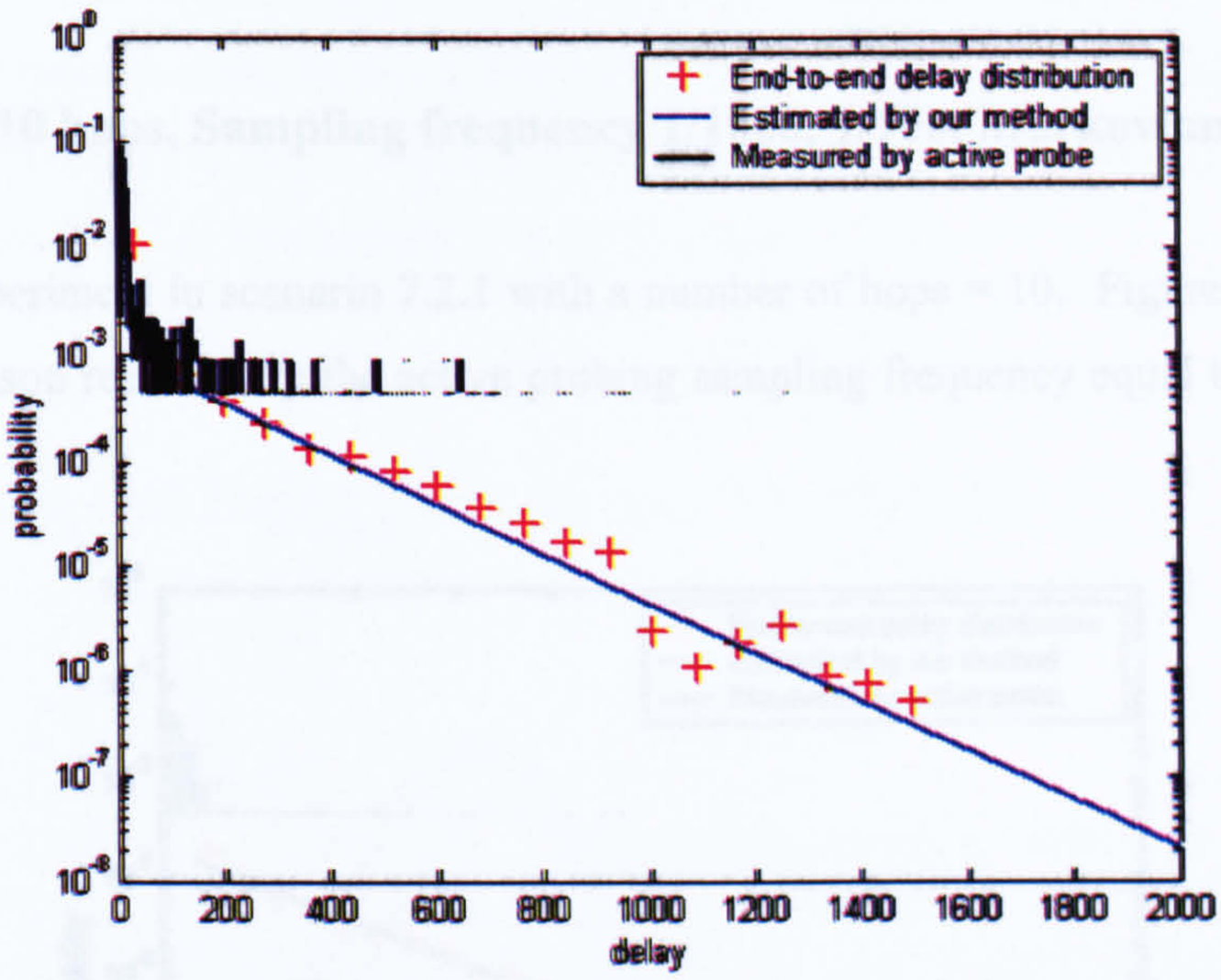


Figure 7-7 End-to-end delay distribution comparison (active probing sampling frequency= 1/100)

Figure 7-6 and Figure 7-7 show the comparison results for the cases of active probing sampling frequency = 1/1000 and 1/100 respectively. According to the comparison results, it can be seen that the tail of the end-to-end delay distribution of the foreground traffic (red line) is in accordance with our measurement methods (blue line). However, the results estimated by the active probing (black line) are unable to capture the shape of the end-to-end delay distribution of the foreground traffic, even though the sampling frequency is increased to 1/100. Due to the operational cost, in practise, it is expected the active probing traffic should be limited to a small portion of the traffic of interest. It is unaffordable to have comparable testing packet volume with its measured traffic's.

With reference to Figure 7-7, the tail distribution sampled by active probing consists of a train of lines [21]. Compared with the true end-to-end delay distribution, it can be seen that the tail of the true end-to-end delay distribution approximately decays exponentially in this scenario. Clearly, if we are interested in the end-to-end delay distribution, the estimate produced by the active probing does not provide much information about it.

---

[21] With reference to Figure 7-7, it can noted that the distribution measured by active probing is "floored" at about $5 \cdot 10^{-4}$. It implies the number of testing packet generated during measurement period is about $1/(5 \cdot 10^{-4}) = 2000$. In other words, the lowest probability an active probing can be reached is given as $1/N$, where N is number of testing probing injected into the network. The larger the packet delay, the less likely a testing packet experiences. Therefore, the tail of the distribution measured by active probing is discontinuous and few points are displayed. This also demonstrates that a significant amount of testing packets are necessary to capture the distribution.

7.2.2 Scenario: 10 hops, Sampling frequency 1/1000, 1/100, Markovian traffic

We repeat the experiment in scenario 7.2.1 with a number of hops = 10. Figure 7-8 and Figure 7-9 show the comparison results with the active probing sampling frequency equal to 1/1000 and 1/100 respectively.

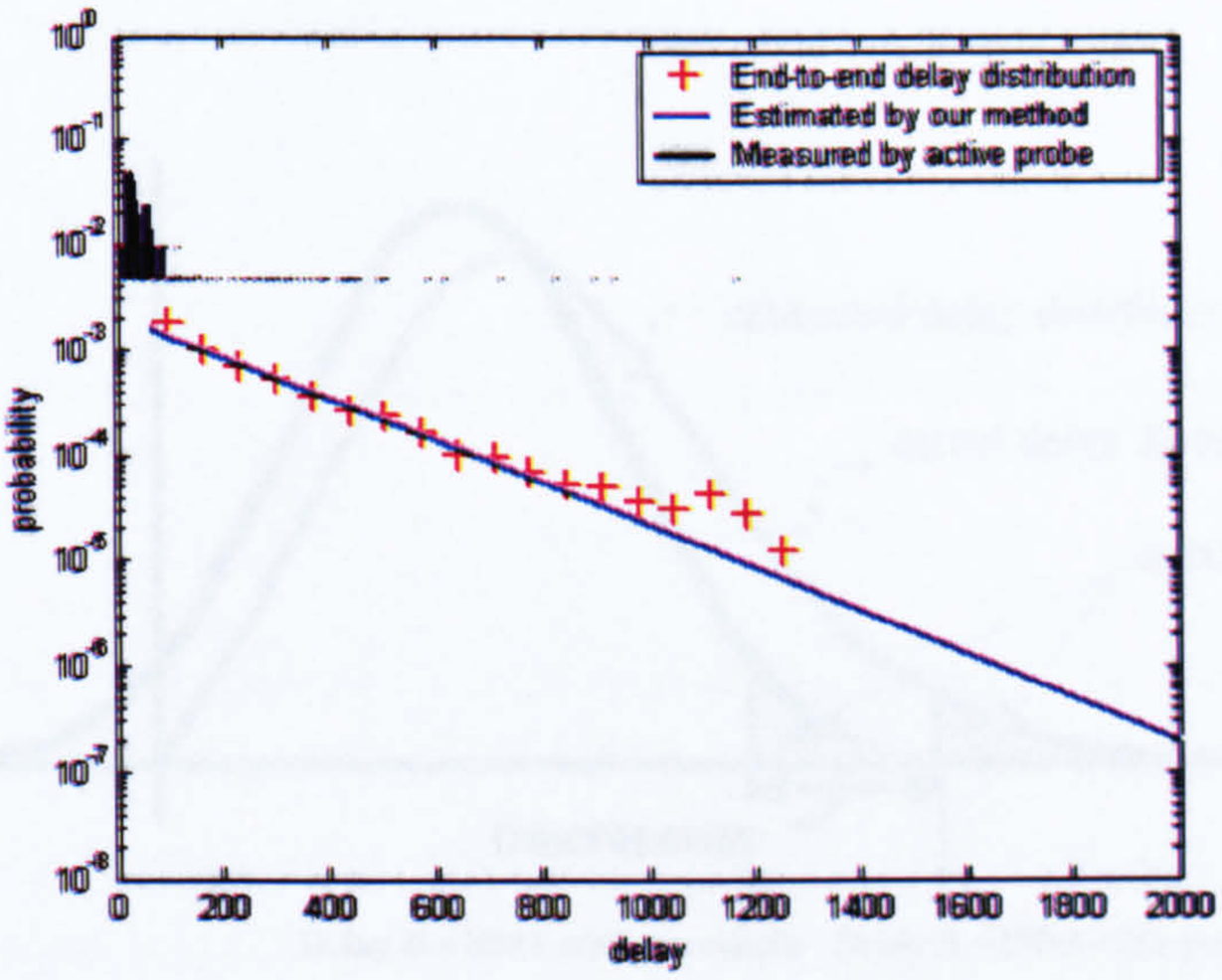


Figure 7-8 End-to-end delay distribution comparison (active probing sampling frequency = 1/1000)

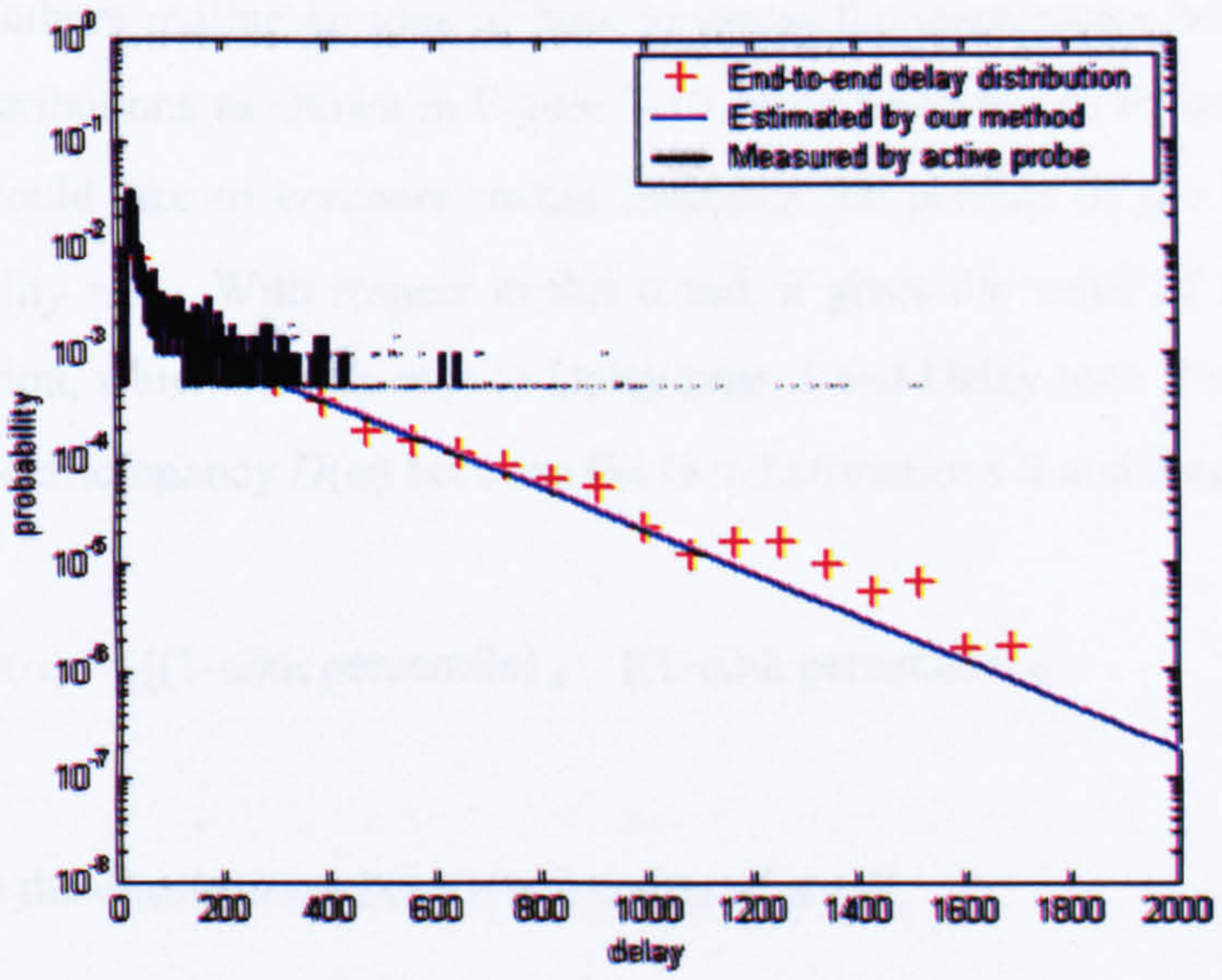


Figure 7-9 End-to-end delay distribution comparison (active probing sampling frequency = 1/100)

With reference to Figure 7-8 and Figure 7-9, similar phenomena are obtained with scenario 7.2.1. The estimate produced by the active probing fails to present the tail of true end-to-end delay distribution. In contrast to it, the end-to-end delay distribution obtained with our end-to-end delay measurement method is closely matched with the true one.

7.3 SLA End-to-end Delay Performance Verification

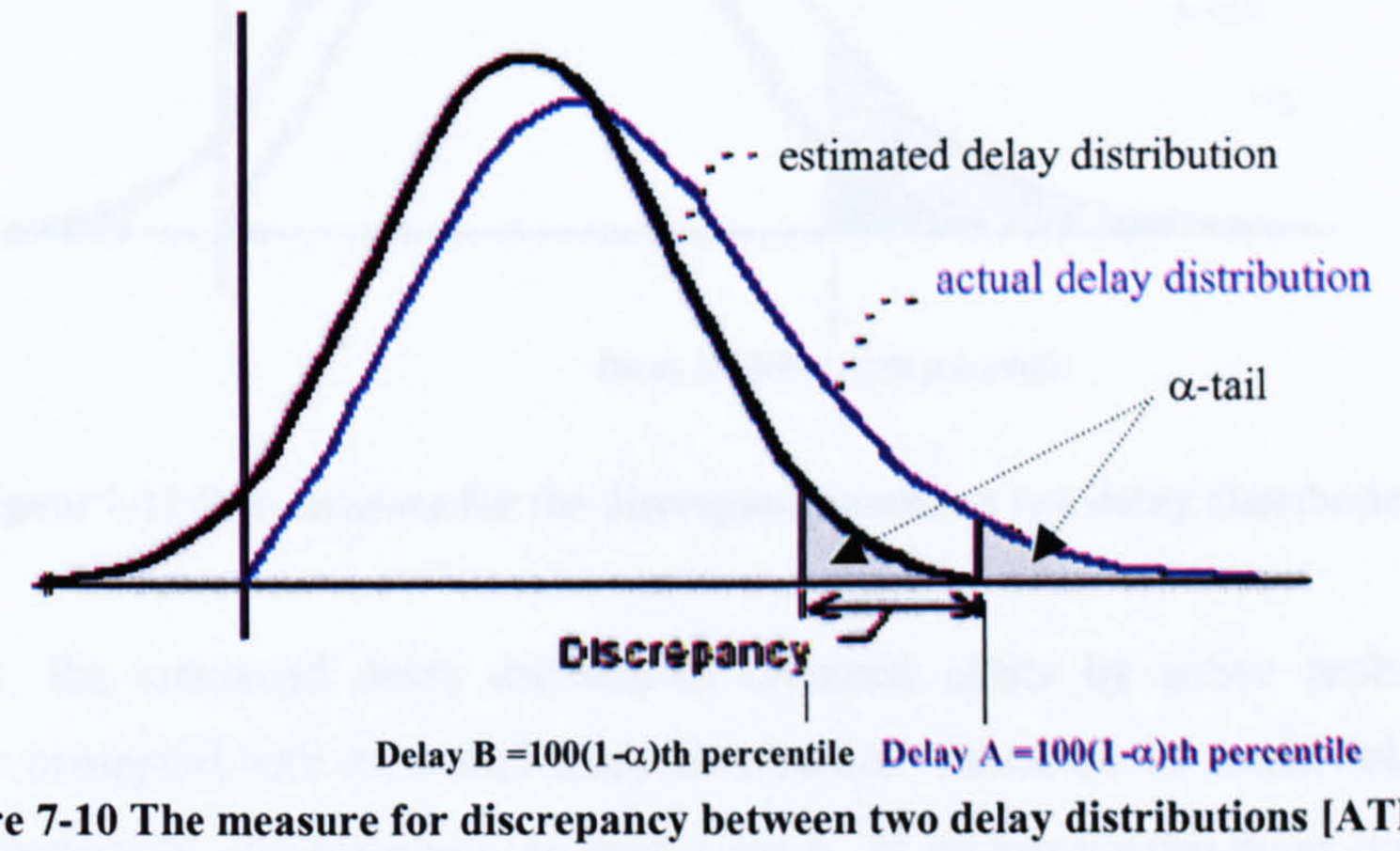


Figure 7-10 The measure for discrepancy between two delay distributions [ATM96]

In [ATM96], the authors outline an idea of how to assess the discrepancy between the actual and estimated delay distributions as shown in Figure 7-10. With reference to Figure 7-10, there are two distributions we would like to compare, α-tail indicates the portion of the tail of a distribution having the probability = α. With respect to this α-tail, it gives the value of 100(1-α)th percentile from each distribution, which corresponds to Delay time *A* and Delay time *B* respectively as shown in Figure 7-10. The discrepancy $D(\alpha)$ between the two distributions *A* and *B* are given as:

$$D(\alpha) = |\{(1-\alpha)\text{th percentile}\}_A - \{(1-\alpha)\text{th percentile}\}_B| \qquad \textbf{Equation 7-1}$$

It can be noted that the discrepancy $D(\alpha)$ is a function of α-tail.

A typical Service Level Agreement normally specifies the requirement in such a way that a specified portion of the transmitted packets should be within a target delay time. Therefore, the α-tail of the delay distribution is sufficient for the verification purpose.

To examine the effectiveness of the performance of our QL Measurement scheme and active probing in SLA latency verification, the approach of determining the discrepancy between two distributions as discussed above is modified as follows:

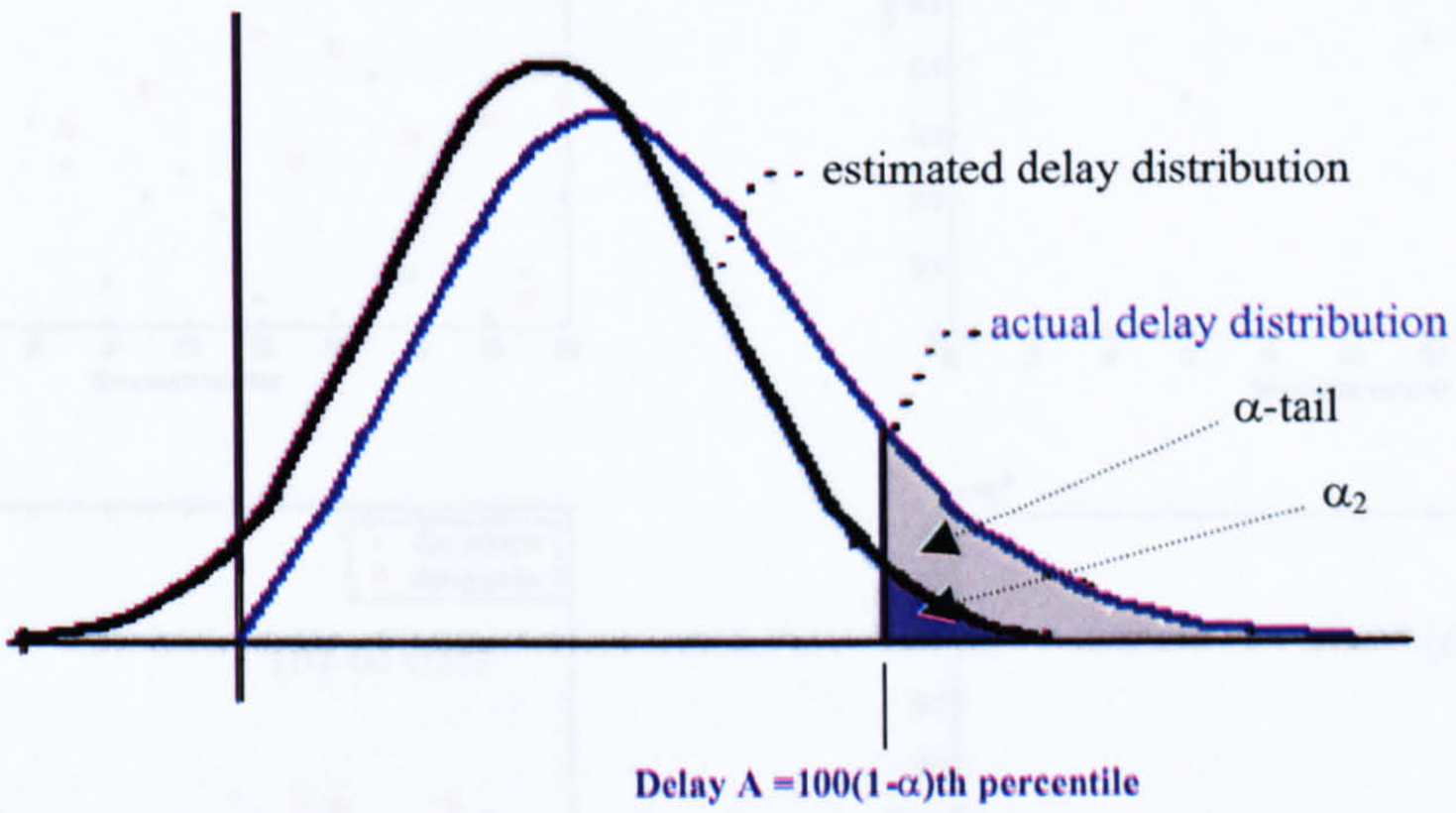


Figure 7-11 Our measure for the discrepancy between two delay distribution

In Figure 7-11, the estimated delay distribution obtained either by active probing or by QL Measurement is compared with the actual delay distribution. Based on the actual delay distribution, the 100(1-α)th percentile of α-tail gives the delay time A. If we assume this delay time as our target latency listed in a SLA, then the tail with respect to this delay time in the estimated distribution estimates the portion of the packets violating the latency requirement. This is given as α_2. Hence, the estimation error is given by the difference between α and α_2. The discrepancy between two distributions is given in Equation 7-2. Similar to Equation 7-1, the discrepancy (estimation error) is a function of α-tail.

$$D(\alpha) = |\alpha - \alpha_2| \qquad \textbf{Equation 7-2}$$

We repeat the experiment with the set-up as in scenario 7.2.1 with active probing frequency 1/1000. The end-to-end distributions are estimated with the QL Measurement scheme and Active Probing. By using Equation 7-2, the estimation error is determined with respect to α-tail = 0.05, 0.01, 0.001 and 0.0001. The experimental result is shown in Figure 7-12.

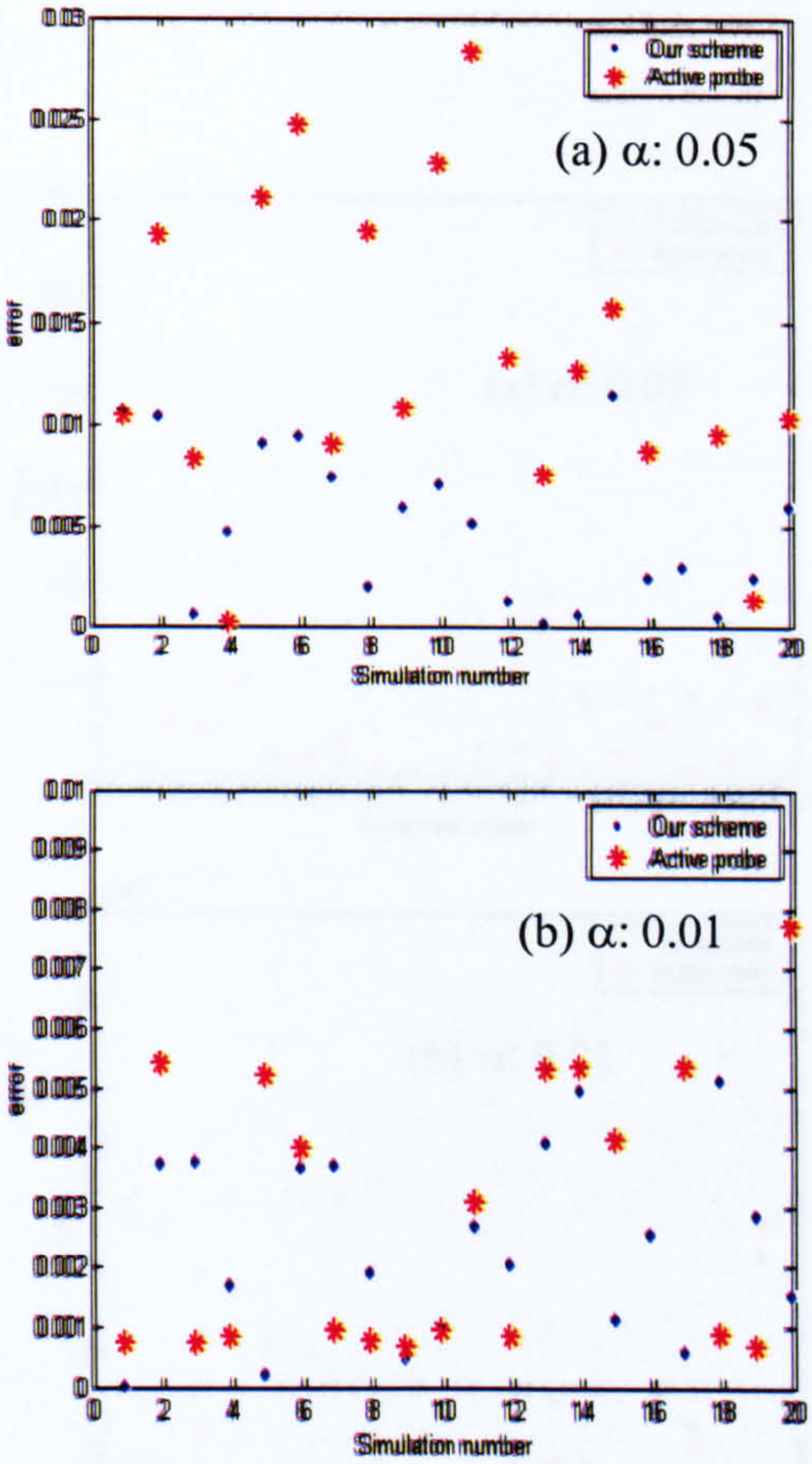


Figure 7-12 Absolute error plot for SLA end-to-end delay verification

Figure 7-12 shows the error plot for various values of α-tail: (a) 0.05, (b) 0.01, (c) 0.001 and (d) 0.0001 for the scenario 7.2.1 with active probing sampling frequency 1/1000. Recall that α-tail means that (1-α) of the foreground traffic packets meets the target value if the end-to-end delay requirement 100(1-α)th percentile is assumed. Clearly, the smaller value of α-tail is, the more stringent the latency requirement is. We performed 20 different experiments to obtain the scatter plot as shown in Figure 7-12. This scatter plot is also used in [ZSE02]. It can be noted that the estimation error with our measurement is within the order of magnitude of α-tail. With the sampling frequency 1/1000, the estimation error with active probing is larger than that with our scheme (see Figure 7-12(a) and Figure 7-12(b)). In addition, for a smaller value of α-tail: 0.001 and 0.0001, the active probing is unable to resolve the figure of the end-to-end packet delay fractions because of insufficient testing packets[22], while our scheme is still able to produce the result.

---

[22] The sampling frequency is not large enough to make the measured end-to-end delay distribution to reach beyond 100(1-α)th percentile. In other words, it fails to indicate the portion of packets which have not met the target if the target is assumed to be 100(1-α)th percentile. Hence, in Figure 7-12 c and d, the points for the active probe disappear.

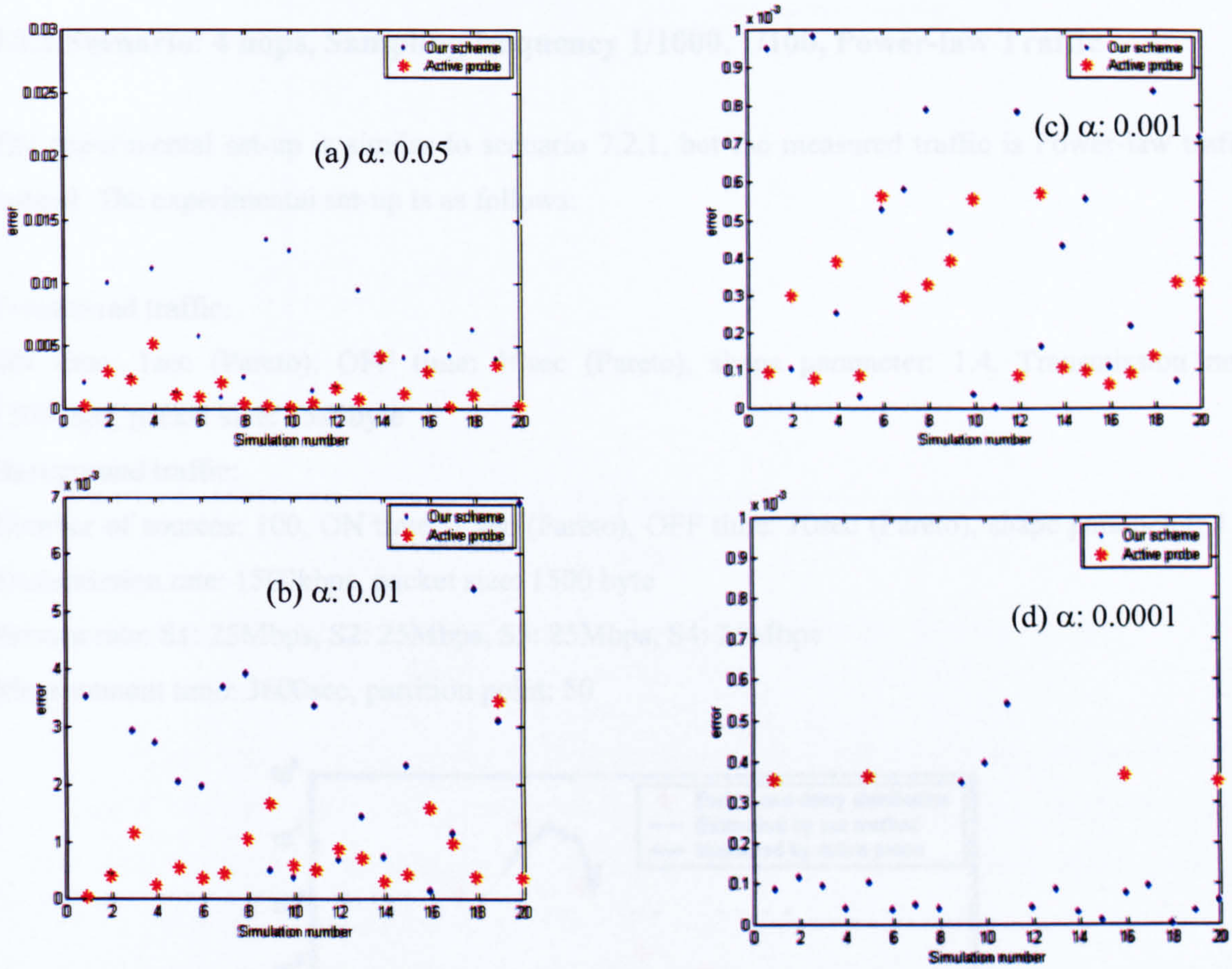


Figure 7-13 Absolute error plot for SLA end-to-end delay verification

Figure 7-13 shows the error plot for various values of α-tail: (a) 0.05, (b) 0.01, (c) 0.001 and (d) 0.0001 for the scenario 7.2.1 with active probing sampling frequency 1/100. Compared with Figure 7-12, it can be noted that the performance of active probing improves as the estimation error is reduced as a whole (see Figure 7-12 and Figure 7-13 (a), (b) and (c)). The result is slightly better than that obtained by the QL Measurement scheme (see Figure 7-13 (a) and (b)). Nevertheless, it still has the problem discussed above such as the active probing is still unable to resolve the figure of the end-to-end packet delay for a smaller value α. With reference to Figure 7-13 (d), only few cases in active probing can give the figure of end-to-end packet delay. This indicates that even if we increase active probing frequency to 1/100, active probing may not be able to provide the answer for a stringent target listed in a SLA.

7.3.1 Scenario: 4 hops, Sampling frequency 1/1000, 1/100, Power-law Traffic

The experimental set-up is similar to scenario 7.2.1, but the measured traffic is Power-law traffic instead. The experimental set-up is as follows:

Foreground traffic:

ON time: 1sec (Pareto), OFF time: 10sec (Pareto), shape parameter: 1.4, Transmission rate: 1500kbps, packet size: 1500byte

Background traffic:

Number of sources: 100, ON time: 1 sec (Pareto), OFF time: 10sec (Pareto), shape parameter: 1.4, Transmission rate: 1500kbps, packet size: 1500 byte

Service rate: S1: 25Mbps, S2: 25Mbps, S3: 25Mbps, S4: 25Mbps

Measurement time: 3600sec, partition point: 50

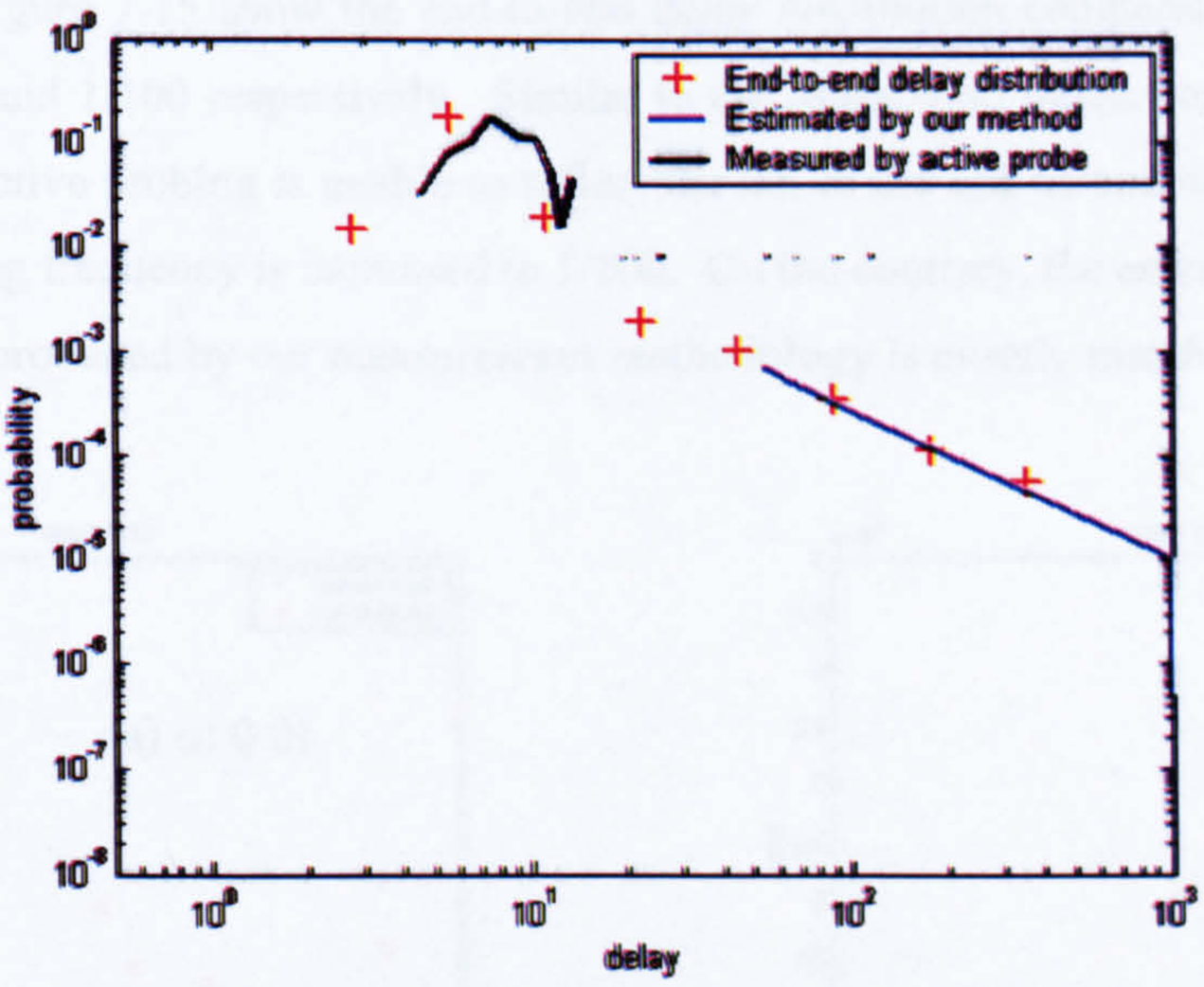


Figure 7-14 End-to-end delay distribution comparison (active probing = /1000)

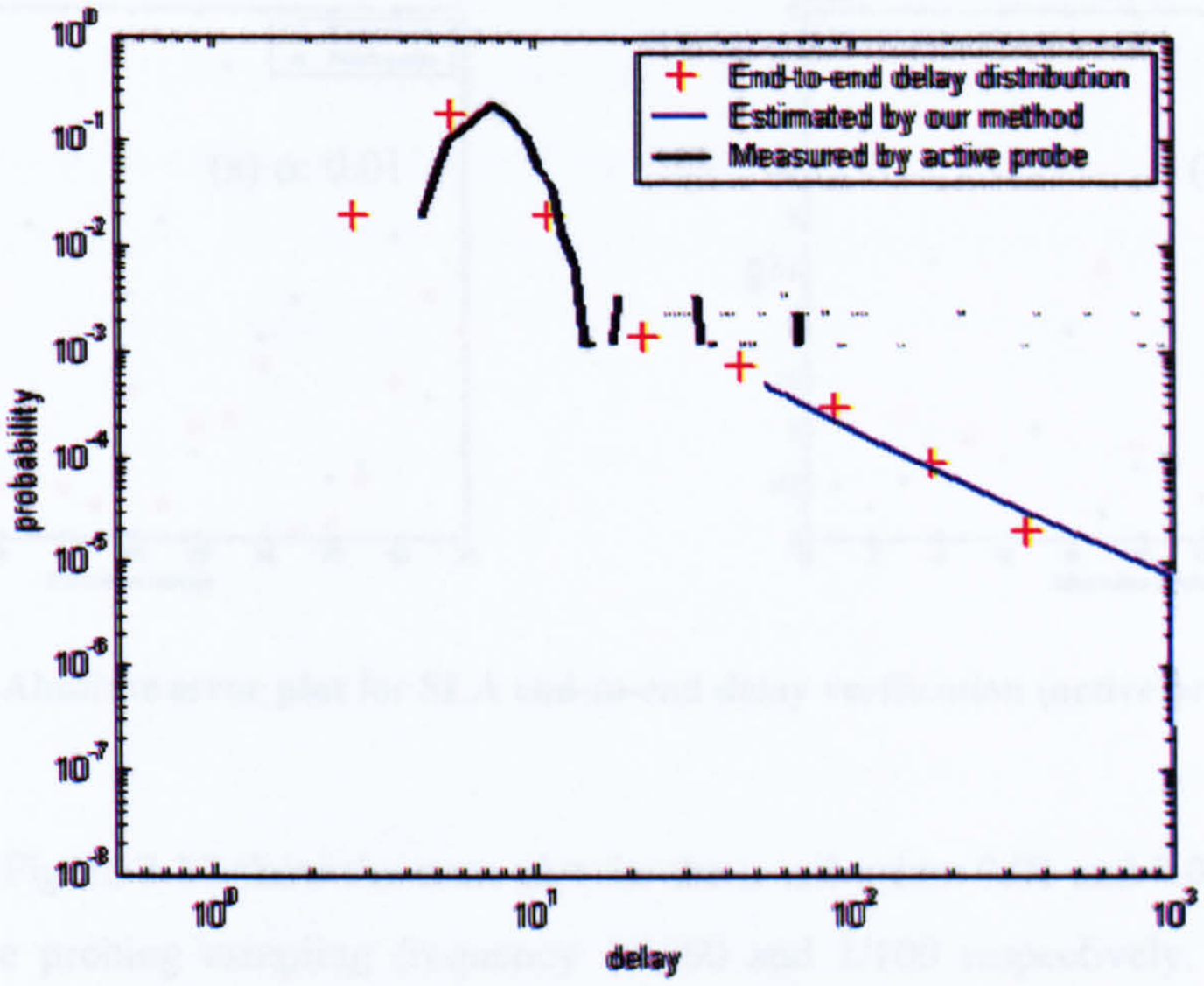


Figure 7-15 End-to-end delay distribution comparison (active probing = 1/100)

Figure 7-14 and Figure 7-15 show the end-to-end delay distribution comparison for active probing frequency 1/1000 and 1/100 respectively. Similar to the Markovian traffic cases, in the Power-law traffic scenarios, active probing is unable to reflect the tail of the end-to-end delay distribution even though the sampling frequency is increased to 1/100. On the contrary, the estimate of the end-to-end delay distribution produced by our measurement methodology is closely matched with the true one.

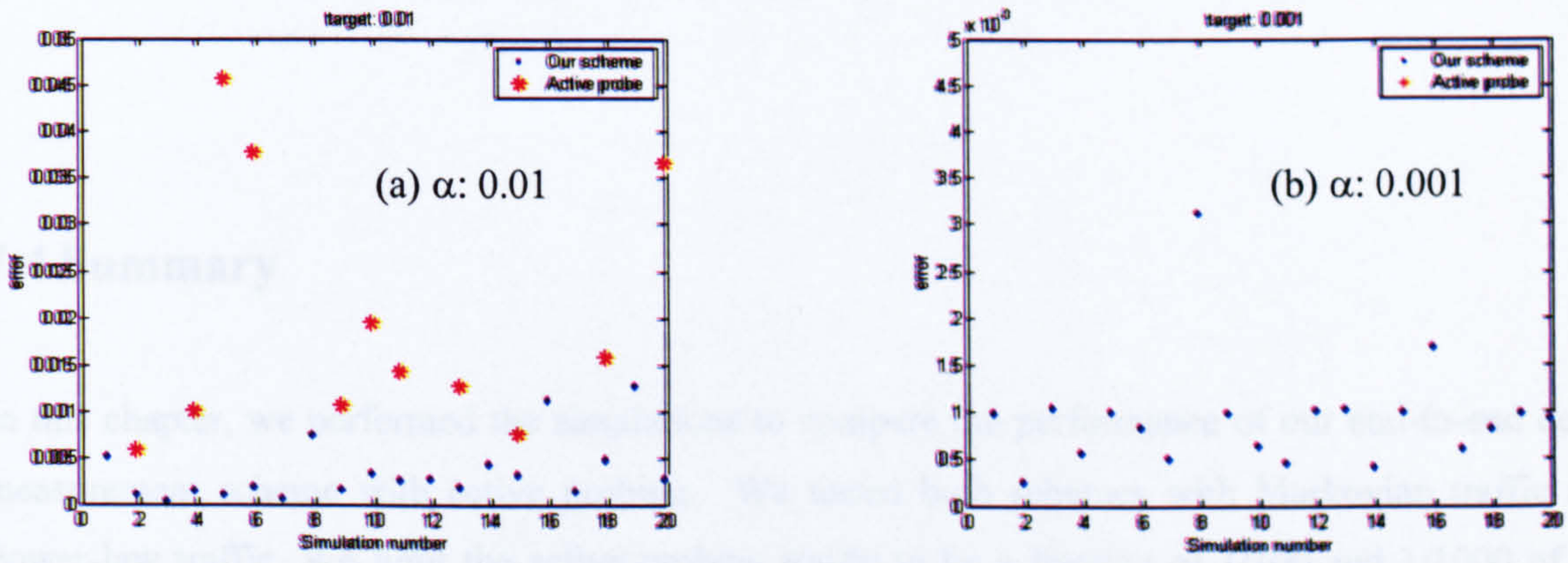


Figure 7-16 Absolute error plot for SLA end-to-end delay verification (active probing = 1/1000)

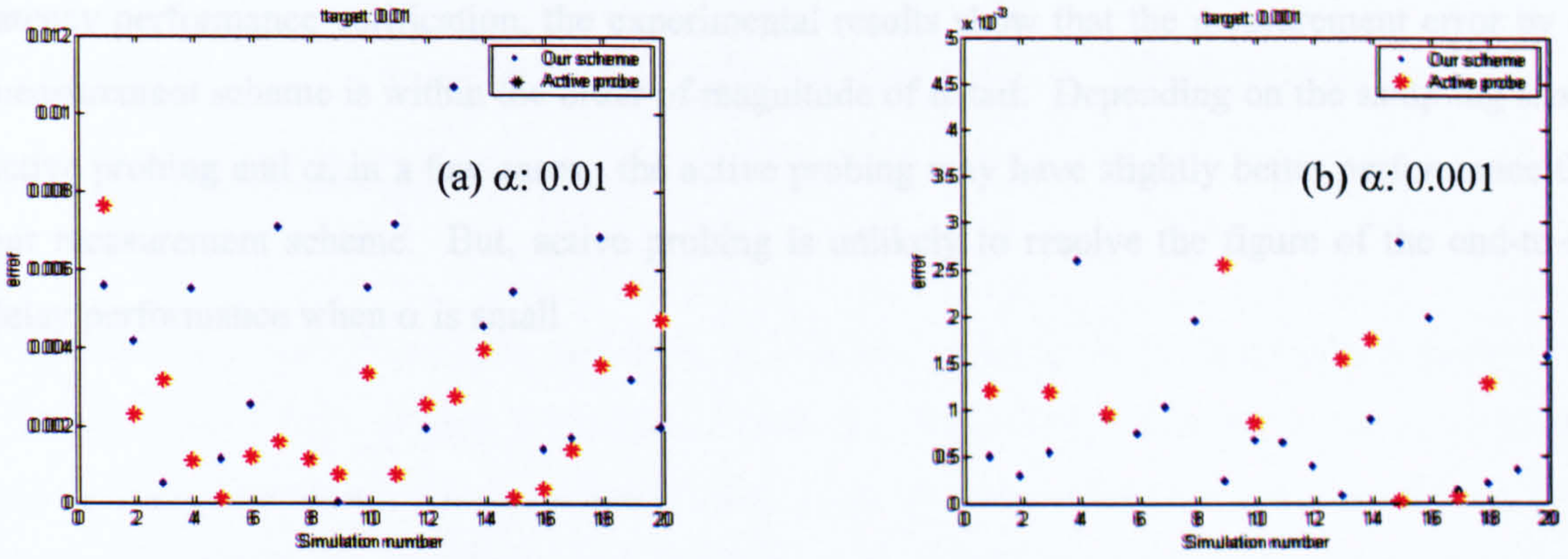


Figure 7-17 Absolute error plot for SLA end-to-end delay verification (active probing = 1/100)

Figure 7-16 and Figure 7-17 show the error plot for the α-tail value: 0.01 and 0.001 for the scenario 7.3.1 with active probing sampling frequency 1/1000 and 1/100 respectively. With reference to Figure 7-16 and Figure 7-17, it can be seen that the measurement error produced by our measurement methodology (blue dots) are still in the same order of magnitude with α. According to Figure 7-16, our end-to-end delay measurement scheme shows better performance than the active probing (with sampling frequency 1/1000) in terms of measurement error. Additionally, when α is small, active probing is unable to resolve the figure of the end-to-end packet delay fractions because of the insufficient testing packets. In Figure 7-17 b, active probing (with sampling frequency 1/100) shows slightly better performance than our measurement scheme in terms of measurement error. However, active probing is again unable to resolve the figure of the end-to-end packet delay fractions even though the sampling frequency is increased from 1/1000 to 1/100.

7.4 Summary

In this chapter, we performed the simulations to compare the performance of our end-to-end delay measurement scheme with active probing. We tested both schemes with Markovian traffic and Power-law traffic. We limit the active probing traffic to be a fraction of 1/100 and 1/1000 of the foreground traffic (the traffic of interest). We assessed both measurement for their effectiveness in capturing the end-to-end delay distribution and the SLA end-to-end delay performance verification. According to the experimental results, the active probing fails to capture the tail of end-to-end delay distribution unless a comparable size of probing traffic with the measured traffic is used. Clearly this will either cause much intrusion into the network or make the measurement costly. For SLA

latency performance verification, the experimental results show that the measurement error by our measurement scheme is within the order of magnitude of α-tail. Depending on the sampling size of active probing and α, in a few cases, the active probing may have slightly better performance than our measurement scheme. But, active probing is unlikely to resolve the figure of the end-to-end delay performance when α is small.

Chapter 8 Further Application of QL Measurement - Bandwidth Adjustment

Apart from the QoS performance evaluation as discussed in the pervious chapters, to provide service with guaranteed QoS is also a challenge. Sufficient bandwidth allocation is essential to meet customers' QoS requirement. Over-provisioning results in under-utilization and may reduce network operator's revenue, on the other hand, insufficient bandwidth causes poor QoS perceived by the customers. In this section, we demonstrate a further feasible application of our QL Measurement scheme. We propose a measurement scheme which incorporates QL Measurement for bandwidth adjustment. Our method combines the analytical model and measurement technique to infer the impact of the changing bandwidth on the QoS performance for Markovian traffic. In this chapter, we illustrate how to carry out the measurements, and perform for the bandwidth adjustment.

8.1 Bandwidth Adjustment

The motivation for bandwidth adjustment with measurement data is addressed in [KES99]. The author points out the necessity of fine-adjustment of bandwidth in order to have optimum network performance. Over-dimensioning results in under-utilization and proportionally reduces the revenue gained by the network operator. On the contrary, insufficient bandwidth allocation causes poor QoS received by the customers. It fails to meet customers' expectation.

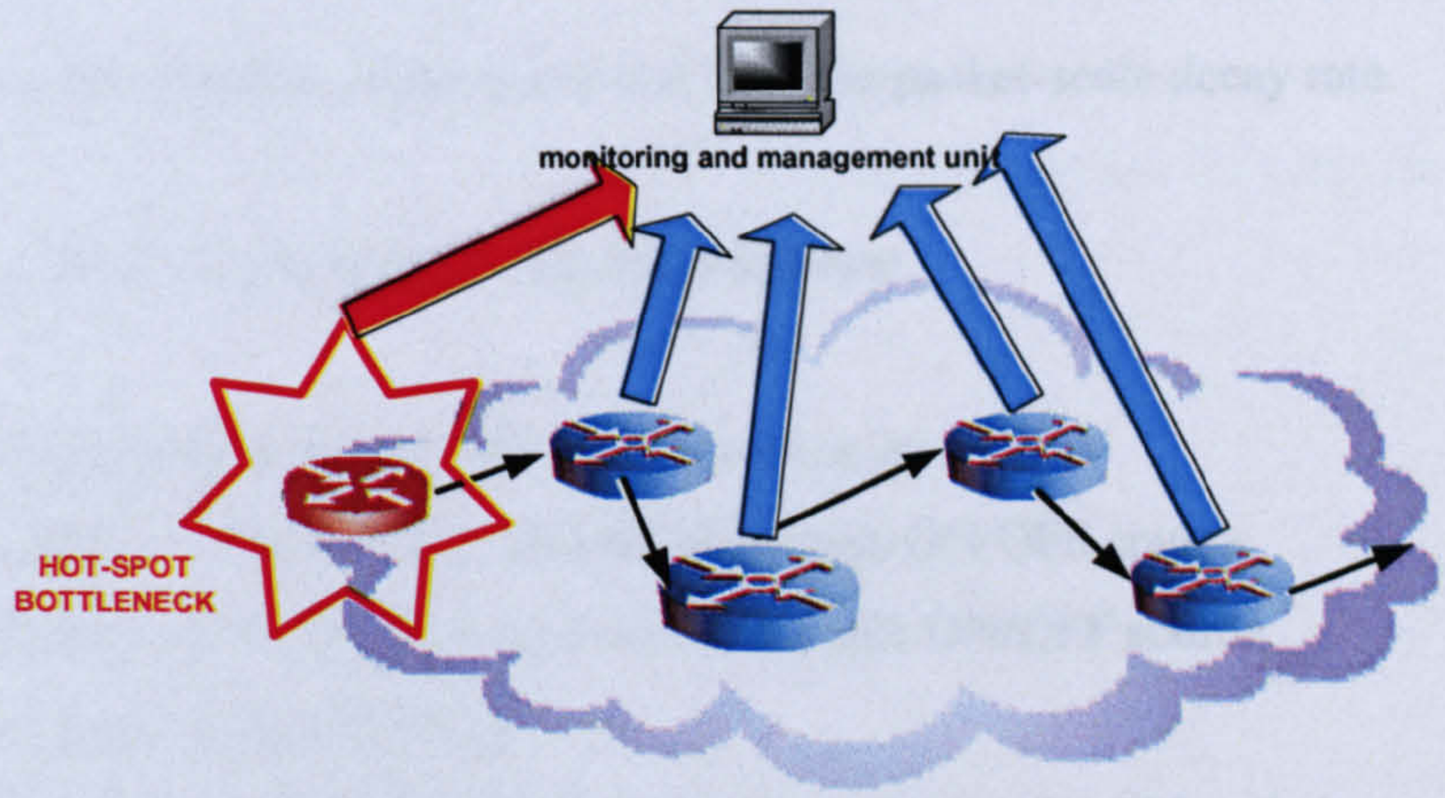


Figure 8-1 The monitoring and management unit identifies the hot-spot

Figure 8-1 depicts the framework for the issue of bandwidth adjustment. The NOC is monitoring the network performance of each node. The local queue length distribution estimated by QL Measurement can be used to infer the local QoS performance for each queue. The hot spot/bottleneck is then identified and located by the management unit. Then, the network operator may wish to allocate more bandwidth to the traffic class in the hotspot. (Here we assume the bandwidth is partitioned by a scheduler mechanism as discussed in Chapter 7.) But there is a question. How much bandwidth is enough?

Recall that wrong aggregation of traffic flows with different statistical features, (for example video and voice traffic), may lead to performance worsening [GAR01]. We assume different traffic types are separated by a scheduler and we are interested in this dimensioning issue for Markovian traffic.

8.2 Making Excess-rate Analysis more Accurate

In Chapter 4, we illustrated the typical queue length distribution characteristic of a FIFO queue multiplexing ON/OFF sources. This consists of two regions often referred to as packet-scale region and the burst-scale region. Both regions can be characterized by decay constant and decay rate. The packet-scale decay rate can be approximated by GAPP analysis [SCH96] as shown in Equation 8-1, whereas, a closed-form formula for the burst-scale decay rate is also available [PIT00].

$$\eta_p = \frac{\rho e^{\rho} - e^{\rho} - \rho^2 + \rho + e^{-\rho}}{\rho - 1 + e^{-\rho}} \qquad \textbf{Equation 8-1}$$

where ρ denotes the utilization of the queue and η_p is the packet-scale decay rate.

The ER theory [PIT00] can be briefly outlined as follows:

Let's define the following symbols with the explanation as:

T_{on} = mean duration of the ON time periods of a single ON/OFF source

T_{off} = mean duration of the OFF time periods of a single ON/OFF source

α = activity factor i.e. $T_{on}/(T_{on}+T_{off})$

R = packet arrival rate of the single ON/OFF source during the ON periods

C = bandwidth allocated to these aggregate ON/OFF sources

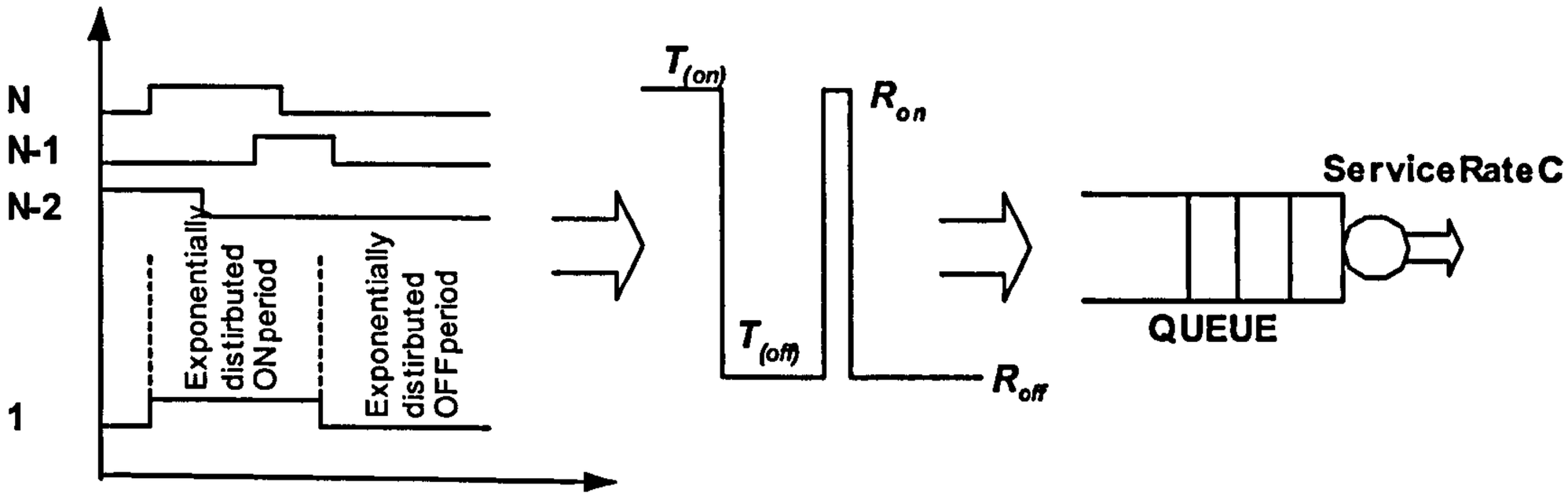


Figure 8-2 State space reduction method [PIT00]

In ER Analysis, N ON/OFF sources are parameterised into a single equivalent ON/OFF source with the parameters $T_{(on)}$, $T_{(off)}$, R_{on} and R_{off} as shown in Figure 8-2. The definitions of these parameters are as follows:

$T_{(on)}$ = mean ON time of the equivalent single ON/OFF source

$T_{(off)}$ = mean OFF time of the equivalent single ON/OFF source

R_{on} = transmission rate of the equivalent single ON/OFF source during ON period

R_{off} = transmission rate of the equivalent single ON/OFF source during OFF period

The ON and OFF time of the single equivalent ON/OFF source are still assumed to be accurately modelled by an exponentially distribution. During the ON period of the equivalent single aggregated source, the transmission rate R_{on} is greater than the service rate. Hence, the queue is built up and causes burst-scale queuing. During an OFF period, the transmission rate R_{off} is smaller than the service rate, and the queue built up during ON periods will be gradually reduced.

The calculation of theses parameters of a single equivalent aggregate ON/OFF source can be found in [PIT00], and the formulae are listed in Table 8-1.

i. $T_{(ON)} = \frac{R \cdot T_{ON}}{C - A_p}$	mean ON time (aggregate model)
ii. $T_{(OFF)} = T_{(ON)} \frac{1-D}{D}$	mean OFF time (aggregate model)
iii. $R_{ON} = C + R \cdot \frac{A_p}{C - A_p}$	mean rate in the ON state (aggregate model)
iv. $R_{OFF} = \frac{A_p - D \cdot R_{ON}}{1-D}$	mean rate in the OFF state (aggregate model)
v. $A_p = F \cdot T_{ON} \cdot R$	mean load in packets/sec
vi. $F = \frac{N}{T_{ON} + T_{OFF}}$	the rate of flow arrivals
vii. $A = F \cdot T_{ON}$	the offered traffic in Erlangs
viii. $N_o = \frac{C}{R}$	minimum number of active sources for burst-scale queuing
ix. $D = \frac{\left\{\frac{A^{N_o}}{N_o!} \cdot \left(\frac{N_o}{N_o - A}\right)\right\}}{\left\{\sum_{r=0}^{N_o-1} \frac{A^r}{r!} + \frac{A^{N_o}}{N_o!}\left(\frac{N_o}{N_o - A}\right)\right\}}$	the probability a flow is delayed

Table 8-1 Formula for burst-scale decay rate

[PIT00] gave an equation for the burst-scale decay rate of N homogeneous ON/OFF sources:

$$\eta_b = \frac{a}{s} \qquad \textbf{Equation 8-2}$$

where:

$$a = 1 - \frac{1}{T_{(on)}(R_{on} - C)} \qquad \textbf{Equation 8-3}$$

$$s = 1 - \frac{1}{T_{(off)}(C - R_{off})} \qquad \textbf{Equation 8-4}$$

Substitute Equation 8-3 and Equation 8-4 into Equation 8-2

$$\eta_b = \frac{1 - \dfrac{1}{T_{(on)}\left(R_{on} - C\right)}}{1 - \dfrac{1}{T_{(off)}\left(C - R_{off}\right)}}$$

Equation 8-5

Equation 8-5 provides a closed-form equation for the estimation of burst-scale decay rate [PIT00]. While researching the work reported in this thesis, it became apparent that this approach fails under certain circumstances of interest here. To provide sufficient accuracy, we therefore proposed and published, an enhanced version which yields a considerable improvement in accuracy in the situations of interest [LEU03]. The conceptual development of this new approach is now outlined.

To deduce the Equation 8-5, it was assumed that burst-scale queuing effect happened when the simultaneous transmission rate is greater than the service rate. In other words, the burst-scale queuing effect happens when the number of connections which are in the ON state are simultaneously greater than $N_o = C/R$ (see Table 8-1). However, owing to the presence of the packet-scale queuing effect, short ON durations of aggregate traffic may not contribute significantly to the burst-scale queuing effect. This phenomenon is illustrated in Figure 8-3.

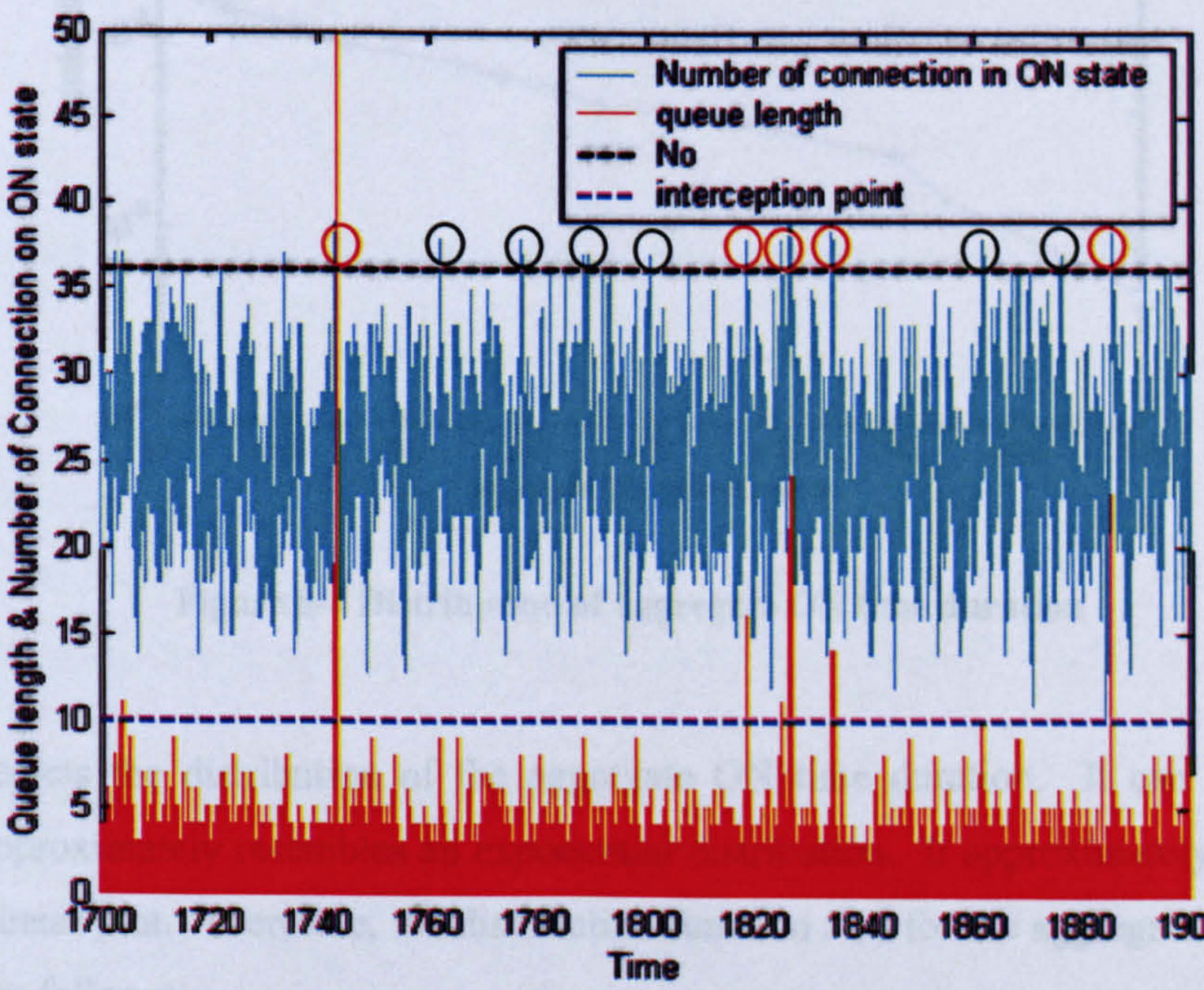


Figure 8-3 Queue length versus the number of connections which are in ON state

Figure 8-3 shows an epoch of the evolution of the queue size and the number of connections which are in the ON state. A FIFO queue served by a rate, C: 6049.8 pkt/sec is multiplexing 70 ON/OFF

sources with transmission rate R = 167 pkt/sec. Therefore, $N_o = C/R$ = 36 that means burst-scale queuing would be expected to take place when more than 36 connections are in the ON state. With reference to Figure 8-3, the cyan line shows the number of connections which are ON, whereas, the red line shows how the queue length evolves with time. Let's assume that the knee point between the packet-scale and burst-scale queuing (see Figure 4-5) is 10 packets as indicated by the blue-dashed line. The circles indicate the event that the number of connections in active state exceeds the threshold N_o. However, the burst-scale queuing effects actually happen in those events circled in red only. Those circled by black circle can not contribute significantly to burst-scale queuing effect, although the number of connections in an ON state is greater than N_o. Therefore, the incoming packets do not contribute to burst-scale queuing but add to packet-scale queuing. This suggests that if the aggregate ON duration is very small, the original formulation of the formula for η_b will tend to underestimate burst-scale queuing. This is what was found.

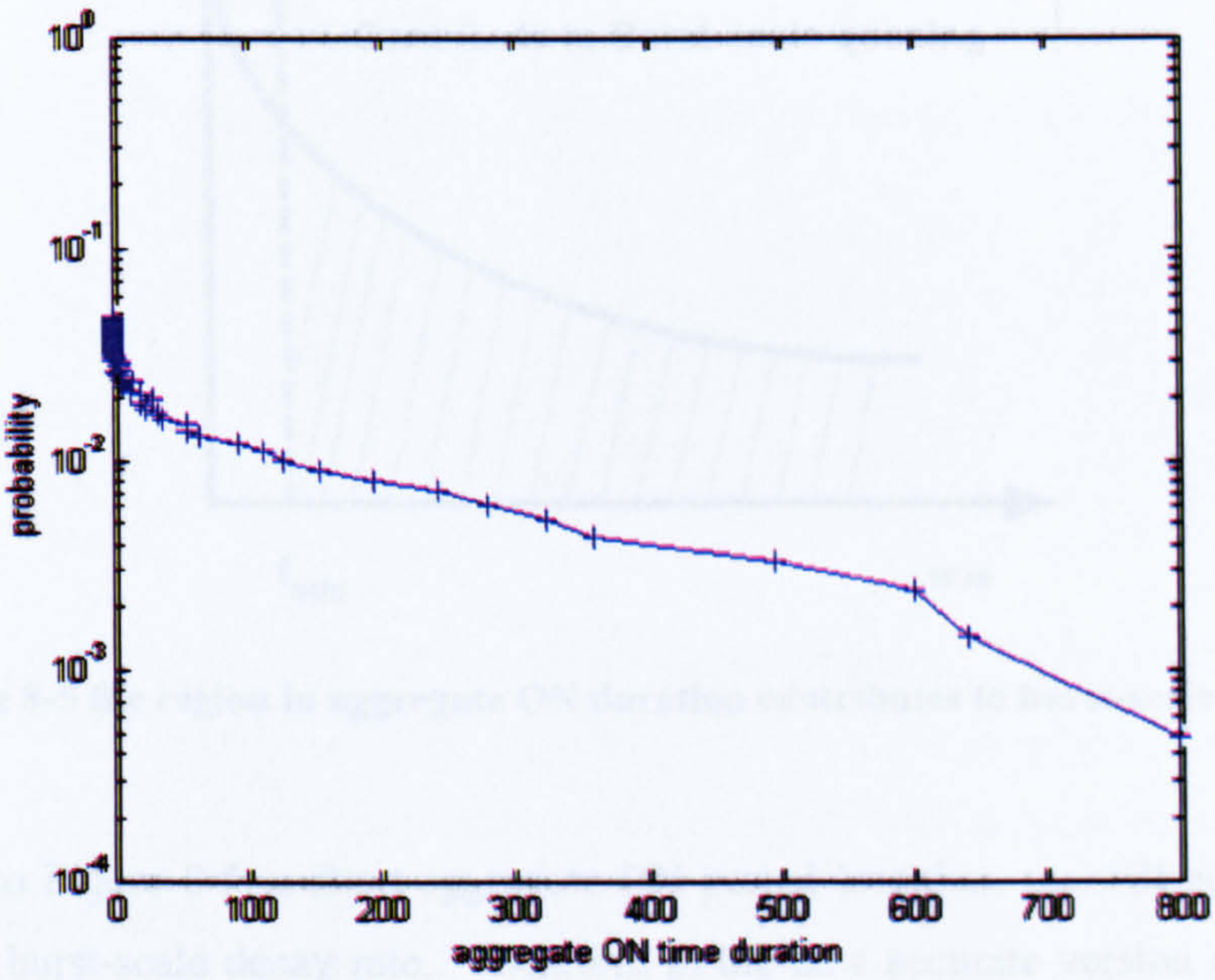


Figure 8-4 Distribution of aggregate ON time duration

Figure 8-4 depicts the distribution of the aggregate ON time duration. It can be seen that the distribution approximately resembles an exponential distribution. It approximately forms a straight line in a log-linear plot. Therefore, the distribution function $F(t)$ for the aggregate ON duration can be described as follows:

$$F(t) = \lambda e^{-\lambda t} \qquad \textbf{Equation 8-6}$$

where $1/\lambda$ = mean of aggregate ON duration which is equal to $T_{(on)}$ as given in Table 8.1.

In the Fluid model and ER Analysis, after an aggregate ON period t, $(R_{on}\text{-}C)t$ ER packets will be added to the queue Let's denote the packet-scale and burst-scale intercepting point as x_{pb}, then the minimum aggregate ON time t_{min} is required so that the excess packets will be included in the burst-scale queuing.

$$t_{\min} = \frac{x_{pb}}{(R_{on} - C)} \qquad \textbf{Equation 8-7}$$

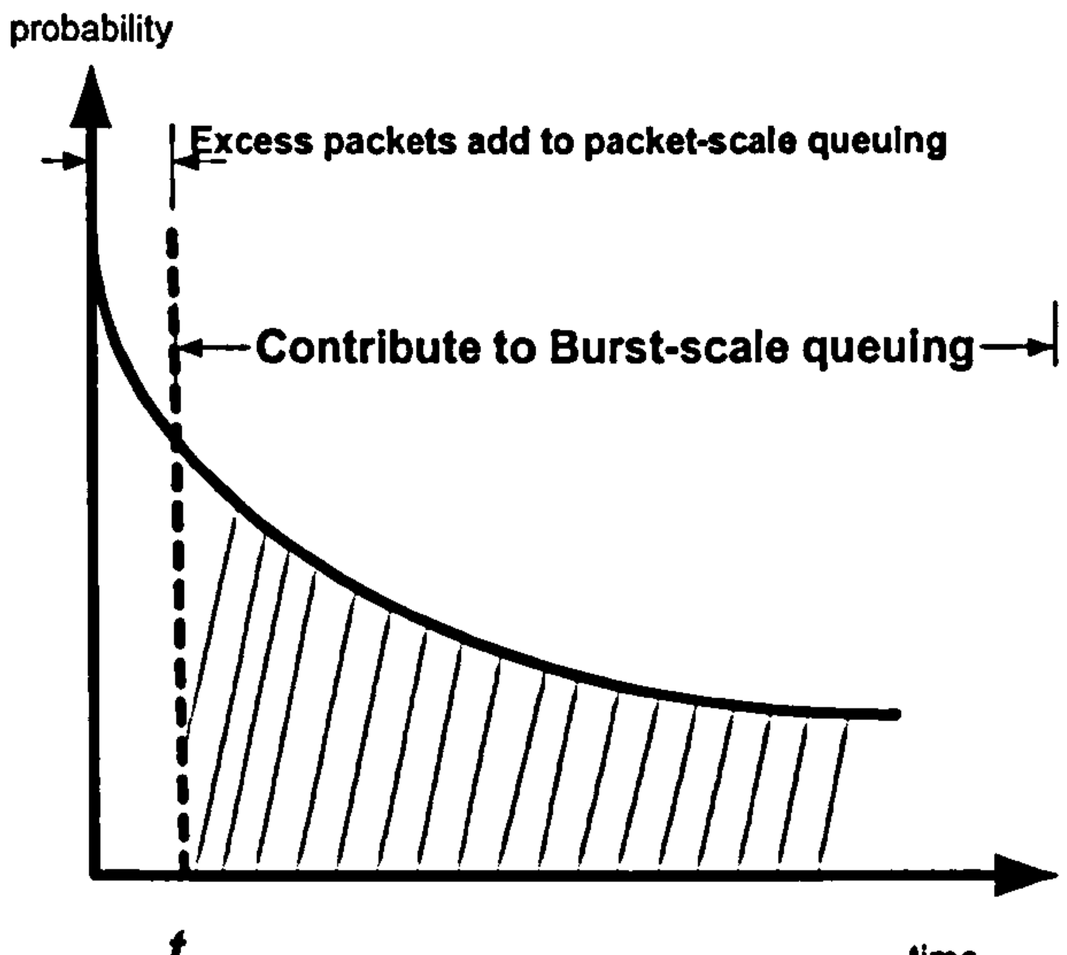


Figure 8-5 the region in aggregate ON duration contributes to burst-scale queuing

With reference to Figure 8-5, a short aggregate ON period less than t_{min} will not be considered in determining the burst-scale decay rate. Therefore in the new accurate version of the analysis, the original mean aggregate ON time $T_{(on)}$ is replaced by a new $T'_{(on)}$ time. This new $T'_{(on)}$ is determined by finding the mean value in the shaded area as in Figure 8-5 as follows:

$$T'_{(on)} = \frac{\int_{t_{\min}}^{\infty} t\lambda e^{-\lambda t} dt}{\int_{t_{\min}}^{\infty} \lambda e^{-\lambda t} dt} = \frac{1}{e^{-\lambda t_{\min}}} \left(\int_{t_{\min}}^{\infty} t\lambda e^{-\lambda t} dt \right) \qquad \textbf{Equation 8-8}$$

Integrating by parts:

$$= \frac{1}{e^{-\lambda t_{\min}}} \left(-te^{-\lambda t} \Big|_{t_{\min}}^{\infty} - \frac{e^{-\lambda t}}{\lambda} \Big|_{t_{\min}}^{\infty} \right)$$

$$= t_{\min} + \frac{1}{\lambda}$$

$$= t_{\min} + T_{(on)} \qquad \textbf{Equation 8-9}$$

Substitute $T'_{(on)}$ in Equation 8-9 into Equation 8-3, then the new modified value a′ will be

$$a' = 1 - \frac{1}{T'_{(on)}(R_{on} - C)}$$

$$= 1 - \frac{1}{(T_{(on)} + t_{\min})(R_{on} - C)}$$

$$= 1 - \frac{1}{x_{pb} + T_{(on)}(R_{on} - C)} \qquad \textbf{Equation 8-10}$$

Substitute a' into Equation 8-2, we obtain new formula for the burst-scale decay rate:

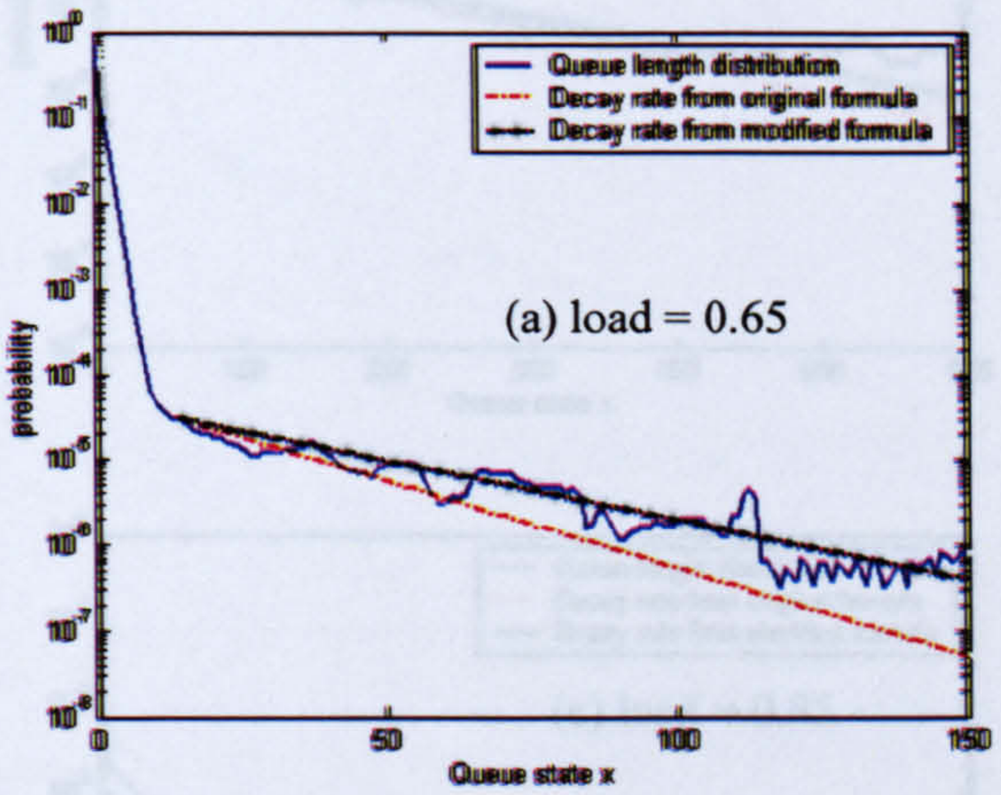
$$\eta_b{}' = \frac{1 - \frac{1}{x_{pb} + T_{(on)}(R_{on} - C)}}{1 - \frac{1}{T_{(off)}(C - R_{off})}} \qquad \textbf{Equation 8-11}$$

Comparing Equation 8-11 with Equation 8-5, the factor of the intercepting point between the packet-scale queuing region and the burst-scale queuing region x_{pb} is introduced into the original formula to improve the accuracy of the equation. Since x_{pb} is a positive real number, therefore, it suggests the original formula may underestimate the actual burst-scale decay rate. This is what was found.

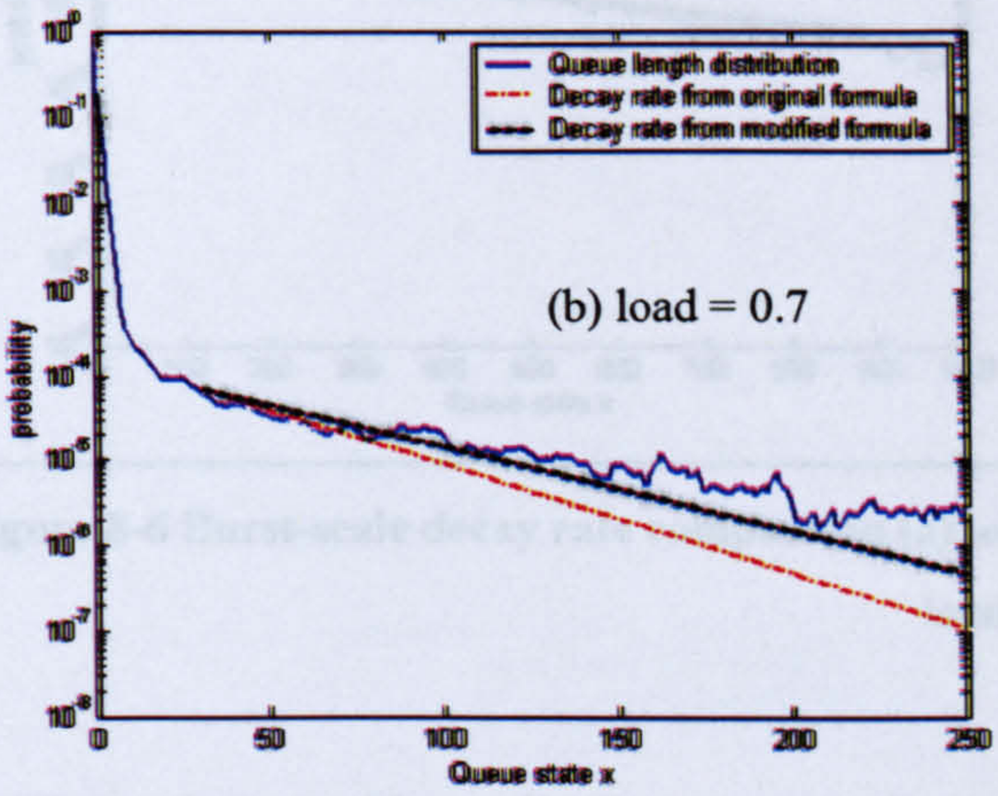
8.2.1 Validation of the New Decay Rate Formula

To validate the accuracy of the new formula Equation 8-11, we performed simulations in which a FIFO queue multiplexed 70 ON/OFF sources. As the cases we studied in the previous chapters, the ON time is 0.96 sec and OFF time is 1.69 sec. To have the burst-scale queuing effect, the load of the queuing system ranges from 0.65 to 0.85. The results are shown in Figure 8-6. With reference to Figure 8-6, we observed that the intercepting point between packet-scale and burst-scale queuing region is around 10 (It is noted that x_{pb} is the interception point between two tangents in the packet and burst-scale region as shown in Figure 4-5, whereas, the partion point q_p is for the measurement

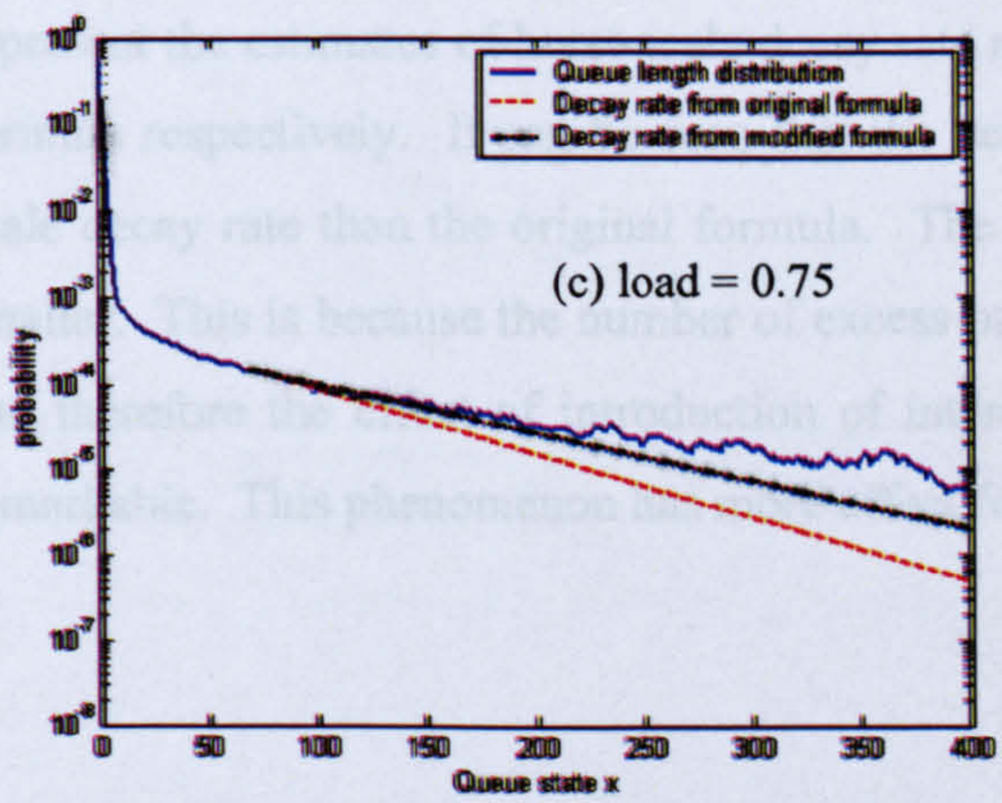
purpose which should be placed within the burst-scale region. The interception point is 10. This also confirms that the selection for q_p should be greater than 20 as discussed in Section 5.3.1). This value is used in the new modified formula as in Equation 8-11.



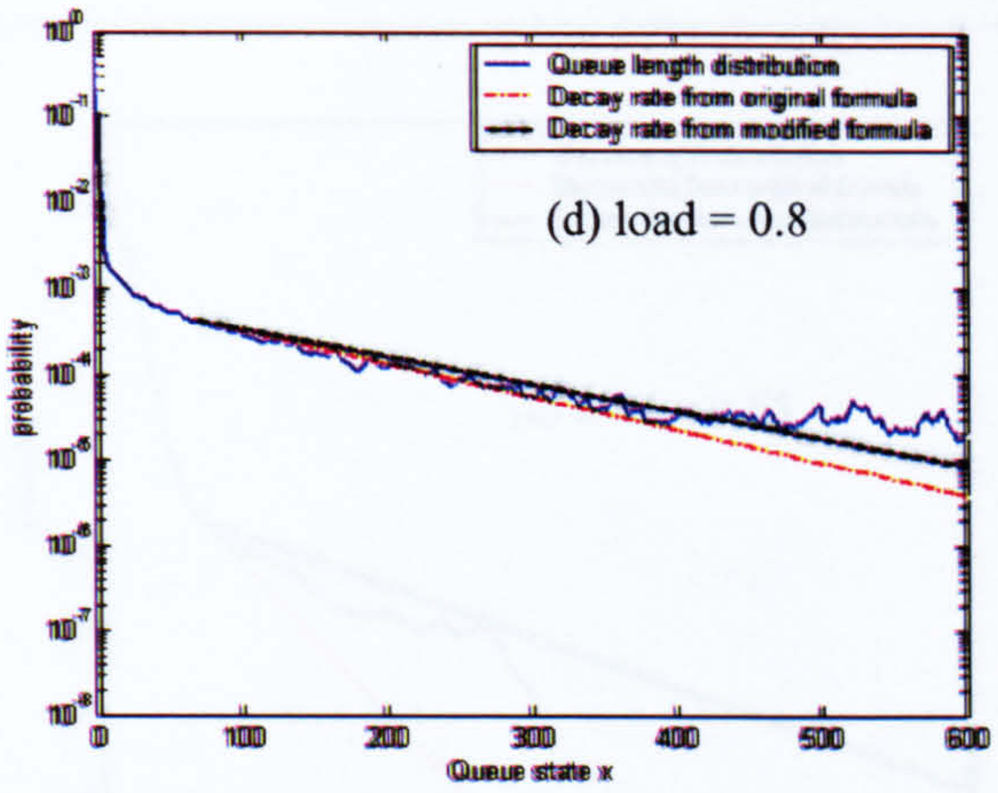
Aggregate parameters	value
R_{on}	$6.8245 \cdot 10^3$
R_{off}	$4.2115 \cdot 10^3$
$T_{(on)}$	0.0702
$T_{(off)}$	8.9934
D	0.0077
A	25.34
a	0.9540
a'	0.9685
s	1.0000



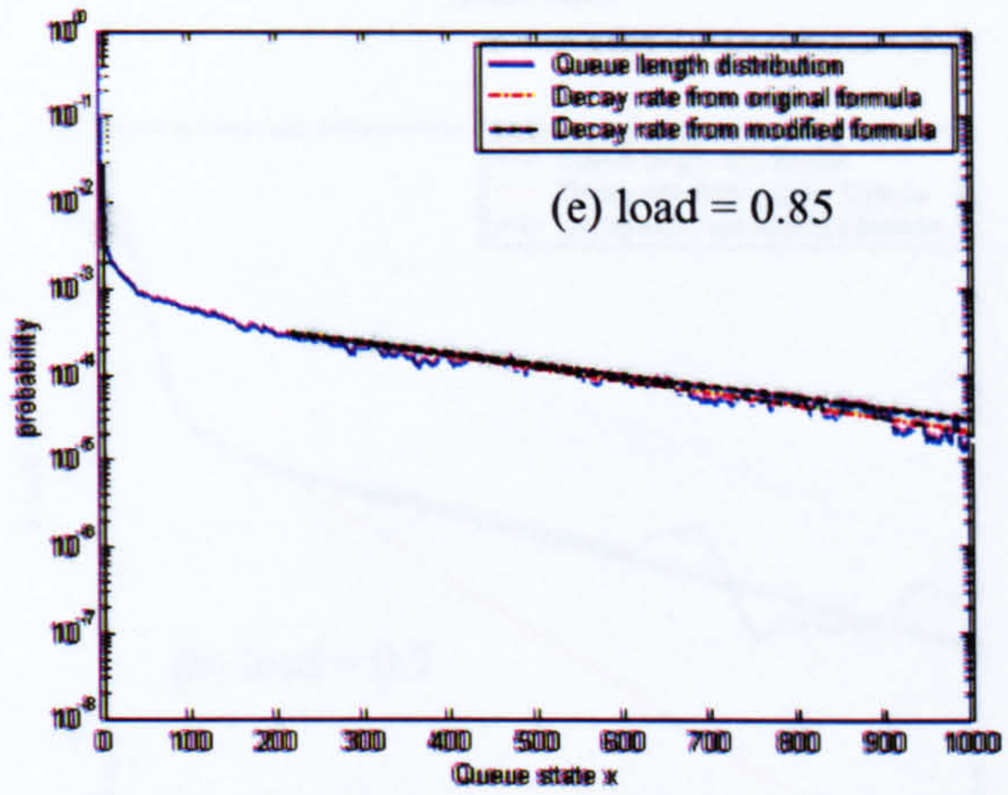
Aggregate parameters	value
R_{on}	$6.4379 \cdot 10^3$
R_{off}	$4.1617 \cdot 10^3$
$T_{(on)}$	0.0882
$T_{(off)}$	2.7787
D	0.0308
A	25.34
a	0.9709
a'	0.9774
s	0.9998



Aggregate parameters	value
R_{on}	$6.1457 \cdot 10^3$
R_{off}	$4.0155 \cdot 10^3$
$T_{(on)}$	0.1134
$T_{(off)}$	1.0030
D	0.1016
A	25.34
a	0.9823
a'	0.9850
s	0.9994



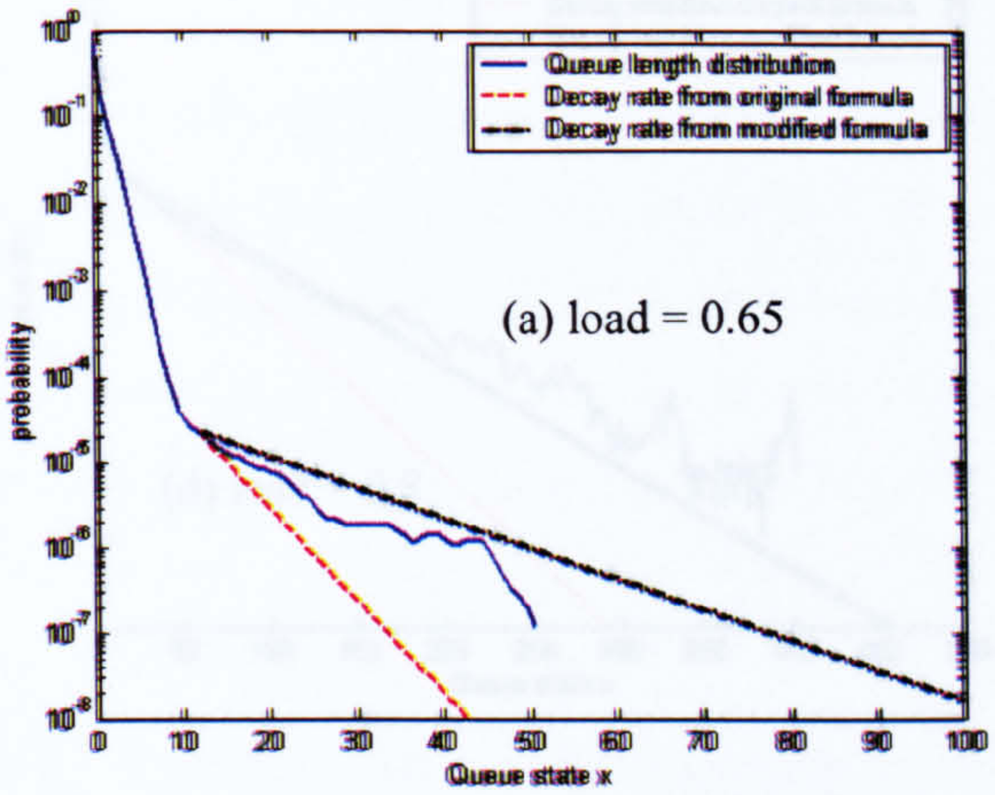
Aggregate parameters	value
R_{on}	$5.9589 \cdot 10^3$
R_{off}	$3.7894 \cdot 10^3$
$T_{(on)}$	0.1511
$T_{(off)}$	0.5898
D	0.2039
A	25.34
a	0.9901
a'	0.9916
s	0.9989



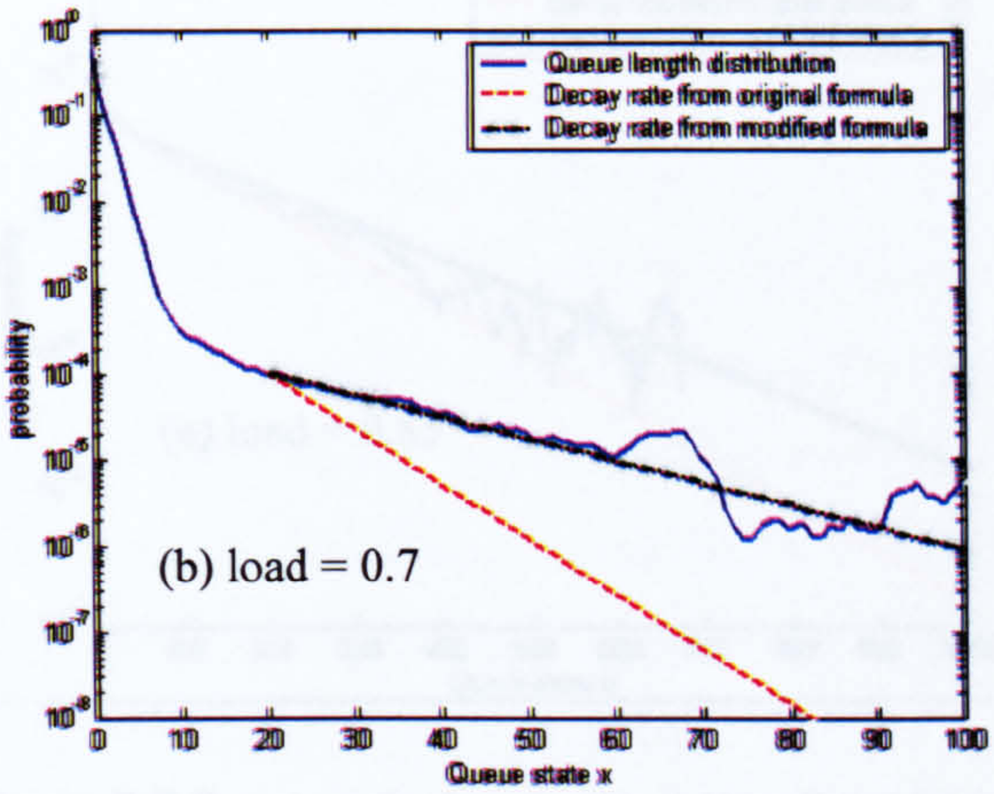
Aggregate parameters	value
R_{on}	$5.9240 \cdot 10^3$
R_{off}	$3.1940 \cdot 10^3$
$T_{(on)}$	0.2137
$T_{(off)}$	0.3485
D	0.3801
A	25.34
a	0.9950
a'	0.9954
s	0.9984

Figure 8-6 Burst-scale decay rate comparison (a) load=0.65, (b) load=0.7, (c) load =0.75, (d) load=0.8, (e) load=0.85

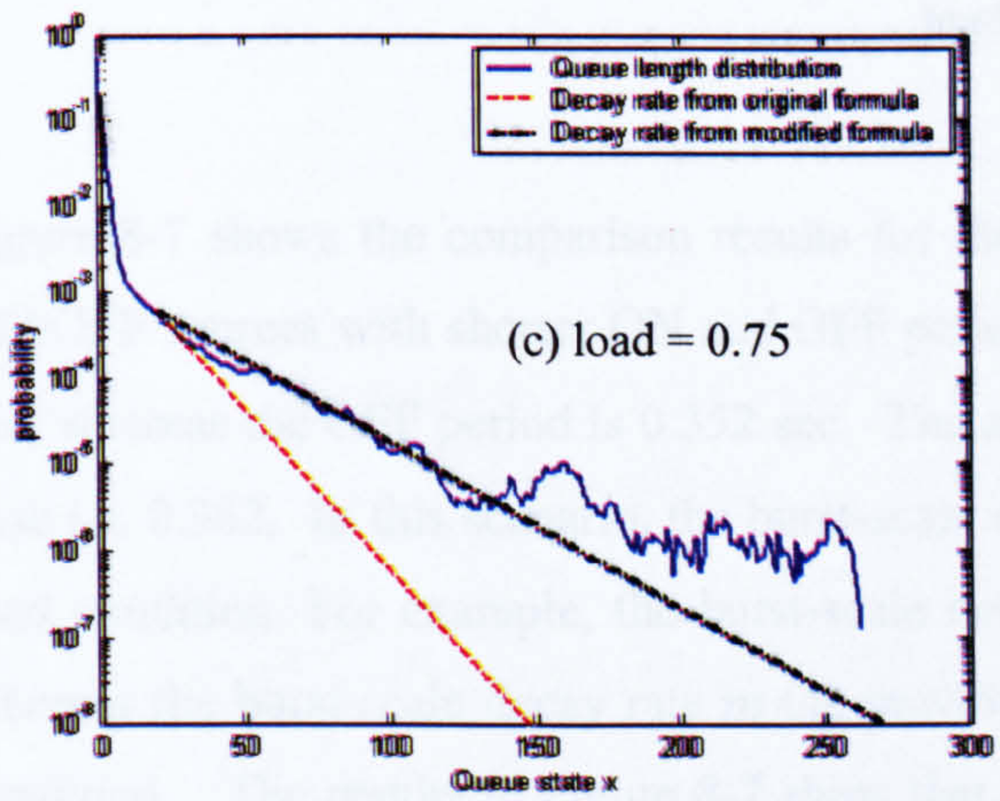
In Figure 8-6, the blue solid lines represent the simulation results, the red and black dashed lines represent the estimates of burst-scale decay rate made by the original formula and the new modified formula respectively. It can be seen that the new formula provides a better estimate of the burst-scale decay rate than the original formula. The improvement is more significant when the load is smaller. This is because the number of excess packets in the aggregate ON period becomes smaller and therefore the effect of introduction of interception point x_{pb} in Equation 8-11 becomes more remarkable. This phenomenon has more effect for shorter ON periods in the ON/OFF sources.



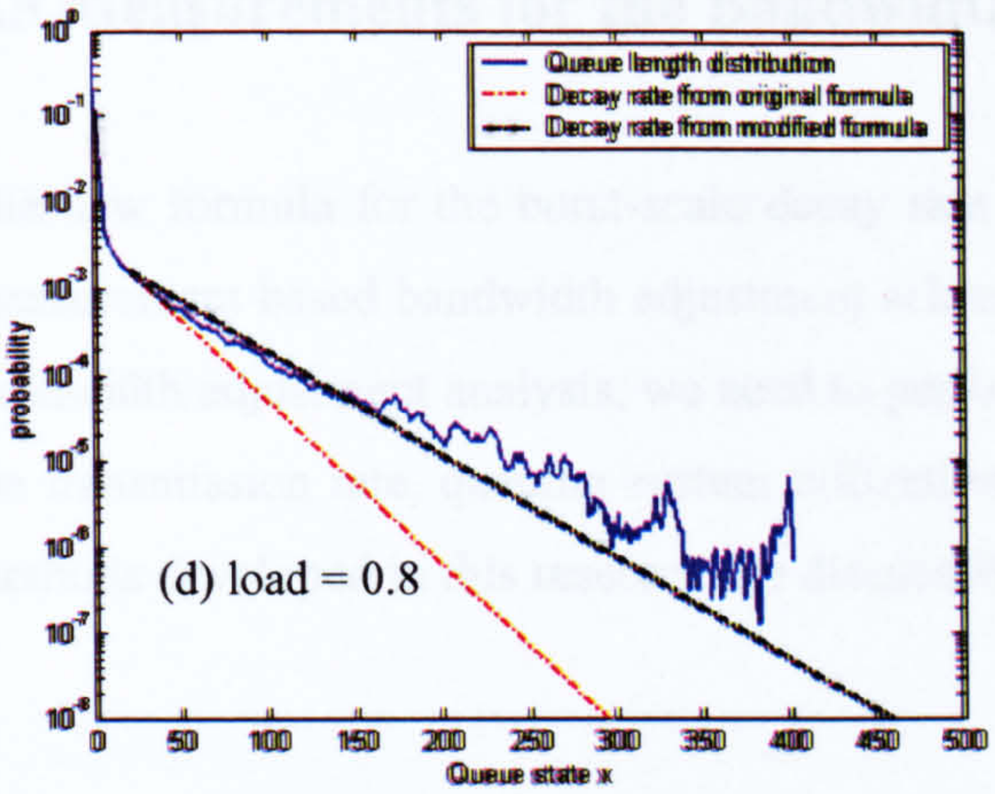
Aggregate parameters	Value
R_{on}	$6.8254 \cdot 10^3$
R_{off}	$4.2115 \cdot 10^3$
$T_{(on)}$	0.0146
$T_{(off)}$	1.8728
D	0.0077
A	25.34
a	0.7790
a'	0.9202
s	0.9998



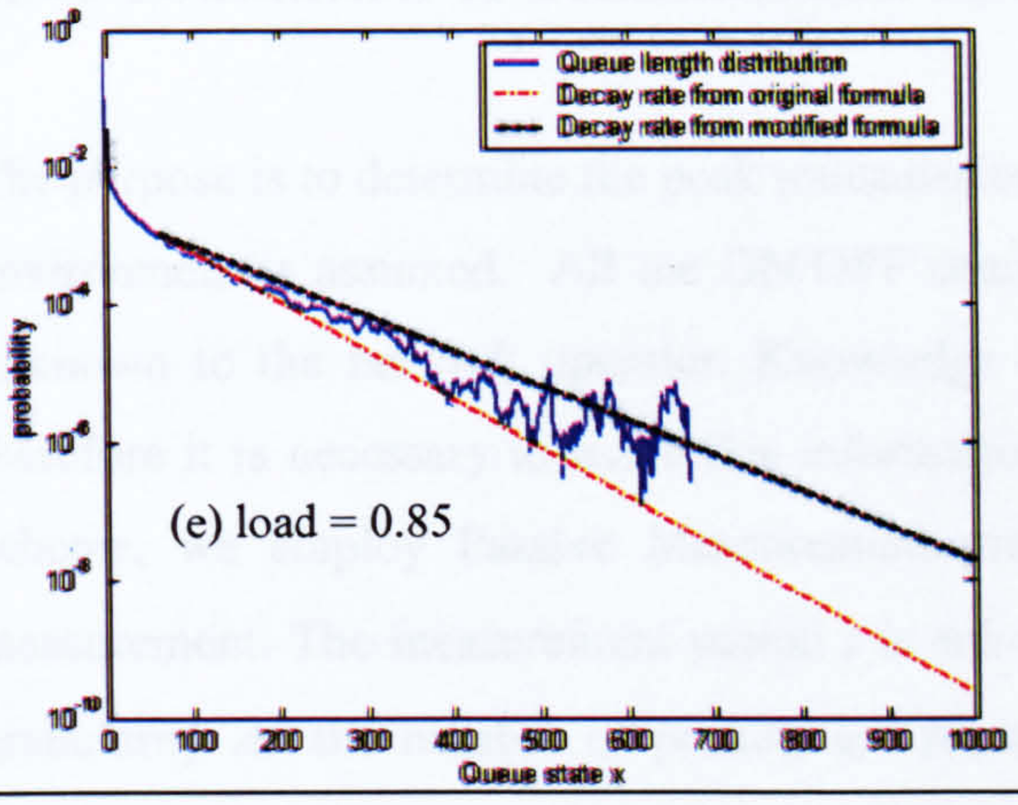
Aggregate parameters	Value
R_{on}	$6.4285 \cdot 10^3$
R_{off}	$4.1620 \cdot 10^3$
$T_{(on)}$	0.0185
$T_{(off)}$	0.5827
D	0.0308
A	25.34
a	0.8619
a'	0.9420
s	0.9991



Aggregate parameters	Value
R_{on}	$6.1464 \cdot 10^3$
R_{off}	$4.0154 \cdot 10^3$
$T_{(on)}$	0.0236
$T_{(off)}$	0.2088
D	0.1016
A	25.34
a	0.9151
a'	0.9541
s	0.9971



Aggregate parameters	Value
R_{on}	$5.9593 \cdot 10^3$
R_{off}	$3.7893 \cdot 10^3$
$T_{(on)}$	0.0314
$T_{(off)}$	0.1228
D	0.2039
A	25.34
a	0.9522
a'	0.9677
s	0.9946



Aggregate parameters	Value
R_{on}	$5.9240 \cdot 10^3$
R_{off}	$3.1940 \cdot 10^3$
$T_{(on)}$	0.0445
$T_{(off)}$	0.0726
D	0.3801
A	25.34
a	0.9762
a'	0.9807
s	0.9923

Figure 8-7 Burst-scale decay rate comparison (a) load=0.65, (b) load=0.7, (c) load=0.75, (d) load=0.8, (e) load=0.85

Figure 8-7 shows the comparison results for the scenario in which a FIFO buffer multiplexes 70 ON/OFF sources with shorter ON and OFF periods. The ON period for the ON/OFF sources is 0.2 sec, whereas the OFF period is 0.352 sec. The activity factor α in this case is same as the previous case i.e. 0.362. In this scenario, the burst-scale decay rate is expected to be smaller under the same load condition. For example, the burst-scale decay rate for the load 0.85 is approximately 0.987, whereas the burst-scale decay rate in the previous case is approximately 0.997 with the same load condition. The results in Figure 8-7 show that the discrepancy between the true burst-scale decay rate with the estimate obtained by the original formula is more significant than the results in Figure 8-6. Again, this discrepancy is reduced by using new modified formula by introduction the factor x_{pb} in the original formula. Note that this could be very important as many bursty applications may have short "active" periods.

8.3 Measurements for the Bandwidth Adjustment Analysis

The new formula for the burst-scale decay rate will be one of the formulas used in our analytical measurement based bandwidth adjustment scheme. To obtain the other relevant information for the bandwidth adjustment analysis, we need to perform the following measurement for the estimation of the transmission rate, queuing system utilization and the burst-scale decay rate. The measurement methods developed in this research are discussed in the following sections.

8.3.1 Estimation of Transmission Rate of ON/OFF source

The purpose is to determine the peak transmission rate R of the ON/OFF source. The homogeneous environment is assumed. All the ON/OFF sources have the fixed transmission rate but these are unknown to the network operator. Knowledge of R is important in the burst-scale analysis and therefore it is necessary to make this information available by measurement. In the measurement scheme, we employ Passive Measurement method (see Section 2.3.2) for the traffic volume measurement. The measurement period t is sub-divided into a smaller granularity Δt. During this granularity Δt, the number of packets are recorded and stored. The probability of a particular ON/OFF source being in ON state is given as its activity factor α, hence, the probability of n connections of N ON/OFF sources being in active state is governed by the binomial distribution i.e.

$$\Pr(number\_of\_active\_connection = n) = {}^{N}C_{n}\alpha^{n}(1-\alpha)^{N-n} \qquad \textbf{Equation 8-12}$$

The mean of a binomial distribution is $N\alpha$.

And its variance is equal to $N\alpha(1-\alpha)$.

The mean number of aggregate incoming packets m in the period Δt is:

$$m = N\alpha R\Delta t \qquad \textbf{Equation 8-13}$$

The variance of the number of aggregate incoming packets v in the period Δt:

$$v = N\alpha(1-\alpha)(R\Delta t)^2 \qquad \textbf{Equation 8-14}$$

During the measurement period t, we will have $t/\Delta t$ samples of the number of incoming packet in the interval Δt. Therefore, we can estimate the mean m and variance v based on these measurement samples using sampling mean and sampling variance respectively.

Rearrange Equation 8-13, the activity factor can be expressed as follows:

$$\alpha = \frac{m}{NR\Delta t} \qquad \textbf{Equation 8-15}$$

Substitute Equation 8-15 into Equation 8-14,

$$R = \frac{Nv + m^2}{Nm\Delta t} \qquad \textbf{Equation 8-16}$$

Equation 8-16 shows the formula to determine the transmission rate of the ON/OFF sources based on the measurement data m and v.

8.3.2 Load Measurement

The load ρ can be simply estimated by measuring the number of packets serviced during the measurement period t.

$$\rho = \frac{Number\_of\_packet\_serviced}{C \cdot t} \qquad \textbf{Equation 8-17}$$

8.3.3 Burst-scale Decay Rate Measurement

QL Measurement is employed to estimate the burst-scale decay rate η_b.

8.4 Bandwidth Adjustment Methodology

Table 8.1 shows the formula to calculate the decay rate of the queue length distribution based on the traffic condition and the allocated bandwidth C. (Recall that the queue length distribution provides useful information for QoS inference.) Assuming that the bandwidth is configured to a new value C', the formula in Table 8.1 can also be re-applied by substituting this new bandwidth C' provided that we have the traffic parameters of the ON/OFF source. Our idea is to make use of the "*measurable*" data as developed in Section 8.3.1-8.3.3. i.e. (R, ρ and η_{pb}) to deduce the necessary traffic parameters listed in Table 8.1. Once these traffic parameters are available, then by *re-applying* the formula in Table 8.1, the impact of changing bandwidth can be determined. The procedure will be illustrated in the following sections.

8.4.1 Estimating N_o, D, R_{on}, R_{off}

Recall that the N ON/OFF sources with parameters (R, T_{on}, T_{off}) are parameterised into a single aggregate ON/OFF model with aggregate parameters (R_{on}, R_{off}, $T_{(on)}$ and $T_{(off)}$) (see Figure 8-2). In Table 8.1, there are some intermediate parameters such as N_o, D, A. N_o is equal to $\lfloor C/R \rfloor$ corresponding to the number of active connections in order to achieve the burst-scale queuing effect. D represents an Erlang loss probability [PIT00]. By using the *measurement results* for R and ρ, an estimate of N_o, Ron and D are given as $N_o = C/R$, $A = \rho C/R$ and $R_{on} = C+RA/(N_o-A)$. D is determined as shown in Table 8.1 and finally R_{off} is equal to $(AR-DR_{on})/(1-D)$. After determining R_{on} and R_{off}, only two parameters T_{on} and T_{off} remain.

8.4.2. Estimating Burst-scale and Packet-scale Knee Point x_{pb}

In Section 8.2, we illustrated that the new formula for the burst-scale decay rate provides better estimation than the original formula. To use this new formula, the information of the interception point x_{pb} must be available. In the packet-scale region, the queue length distribution can be represented as $(1-\rho)\eta_p^x$, where η_p denotes the packet-scale decay rate that can be determined from ρ by using the GAPP formula given in Equation 8-1. The burst-scale queue length distribution is given as $c_p\eta_p^x$. If we extrapolate both lines in the packet-scale region and burst-scale region towards the transition region, then the meeting point is x_{pb}.

$$x_{pb} = \exp\left\{\frac{\ln\left(\frac{1-\rho}{c_b}\right)}{\ln\left(\frac{\eta_b}{\eta_p}\right)}\right\}$$

Equation 8-18

8.4.3 Estimating the Traffic Parameters $T_{(on)}$ and $T_{(off)}$

η_b obtained by QL measurement is now used to determine $T_{(on)}$ and $T_{(off)}$. After this step, we will have all the parameters for the formula with new allocated bandwidth C'. By using the new modified burst-scale decay rate formula Equation 8-11, we get:

$$\eta_b = \frac{1-\frac{1}{T_{(on)}\cdot(R_{on}-C)+x_{pb}}}{1-\frac{1}{T_{(off)}\cdot(C-R_{off})}} = \frac{1-\frac{1}{T_{(on)}\cdot(R_{on}-C)+x_{pb}}}{1-\frac{D}{T_{(on)}(1-D)(C-R_{off})}} = \frac{1-\frac{1}{T_{(on)}H+x_{pb}}}{1-\frac{1}{T_{(on)}G}}$$

Equation 8-19

where $H = (R_{on}\text{-}C)$ and $G = (1\text{-}D)(C\text{-}R_{off})/D$. R_{on}, R_{off}, D and x_{pb} are determined by the measurable data and so H and G.

By re-arranging Equation 8-19, we obtain the quadratic equation.

$$(1-\eta_b)H\cdot G\cdot T_{(on)}^2 + \{(1-\eta_b)x_{pb}G - G + \eta_b H\}T_{(on)} + x_{pb}\eta_b = 0$$

Equation 8-20

$T_{(on)}$ is determined by finding the larger root in the Equation 8-20. The larger root guarantees the positive value of solution for the positive value of $T_{(on)}$. Therefore,

$$T_{(on)} = \frac{-\{(1-\eta_b)x_{pb}\cdot G - G + \eta_b H\} + \sqrt{\{(1-\eta_b)x_{pb}\cdot G - G + \eta_b H\}^2 - 4(1-\eta_b)HG\cdot x_{pb}\cdot\eta_b}}{2(1-\eta_b)HG}$$

Equation 8-21

By using Equation 8-21, the parameter $T_{(on)}$ is determined by the measurable data. Then the parameter T_{on} of any individual ON/OFF source is obtained as $T_{on} = (C\text{-}Ap)T_{(on)}/R$ and $\alpha=\rho C/(NR)$,

T_{off}=(1/α-1). This completes the traffic parameter estimation process i.e. we obtain the necessary parameter R, T_{on}, T_{off} for the next process "Bandwidth Adjustment Estimation".

8.4.4 Bandwidth Adjustment Estimation

In sections 8.4.1 – 8.4.3, we discussed the procedure to determine the traffic parameters R, T_{on} and T_{off} by using measurable data. As discussed above, these parameters are applied to the analytical model in Table 8.1 to determine the impact of changing bandwidth on QoS performance. According to Table 8.1, the parameters D, R_{on}, R_{off}, $T_{(on)}$ and $T_{(off)}$ will change to D', R_{on}', $Roff'$, $T'_{(on)}$ and $T'_{(off)}$ accordingly owing to changing the allocated bandwidth from C to C'. Consequently, the burst-scale decay rate will become as follows:

$$\eta_b{}' = \frac{a'}{s'} = \frac{1-\dfrac{1}{T_{(on)}{}' \cdot \left(R_{on}{}' - C'\right)+x_{pb}}}{1-\dfrac{1}{T_{(off)}{}' \cdot \left(C' - R_{off}{}'\right)}}$$

Equation 8-22

It is noted that the old packet-scale and burst-scale interception point is used in Equation 8-22. This is reasonable as the point does not show much change when the utilization changes. For the burst-scale constant, since the probability for an ER arrival packet is equal to $RD/(C'-Ap)$ [PIT00], which should be equal to the sum of the probability in the burst-scale region. Hence, the burst-scale constant can be expressed as follows:

$$c_b{}' = \frac{R \cdot D'}{C' - A_p}\left(\frac{1-\eta_b{}'}{\eta_b{}'^{x_{pb}}}\right)$$

Equation 8-23

Equation 8-22 and Equation 8-23 determine the burst-scale constant and decay rate with respect to the new allocated bandwidth C'. With these two parameters, the tail of the queue length distribution can be evaluated.

8.5 Experimental Validation

The effectiveness of this measurement-based bandwidth adjustment algorithm is validated through simulation. In the simulation, there are 70 ON/OFF sources with traffic parameters such as mean ON time T_{on} = 0.3sec, mean OFF time T_{off} = 0.528sec (both with exponential distribution), and so the activity factor α = 0.362 and the transmission rate R = 167 packets/sec. These traffic parameters are supposedly not known by the network operator but will be determined by measurement. The original allocated bandwidth is 4235.5 packets/sec for utilization = 0.85. We perform the measurement as discussed in Section 8.3.1 – 8.3.3. The measurement period t is 1000sec with the interval Δt = 50ms. The measurement result is shown as follows:

a. Traffic measurement:

mean number of incoming packets m over Δt : 211.0267

variance of number of incoming packets v over Δt : 1153.8

b. Load measurement:

number of packets received during measurement period: 4200547 packets (Service rate C = 4982.195 packets/sec), therefore, the load is measured to be 0.843

c. Burst-scale delay rate and constant measurement:

$\bar{q}_{high}$ = 220.3823, $\bar{q}_{low}$ = 8, p_{high} = 0.054, p_{low} = 0.946

From Equation 4-15, the burst-scale decay rate $\eta_b = 1\text{-}1/(\bar{q}_{high}\text{-}q_p) = 0.9917$ and the burst-scale decay constant $c_b = \{(1\text{-}\eta_b)p_{high}\}/(\eta_b^{q_p+1}) = 0.001$

Comparison between the true queue length distribution and the estimated distribution with estimated decay rate and decay constant is shown in Figure 8-8.

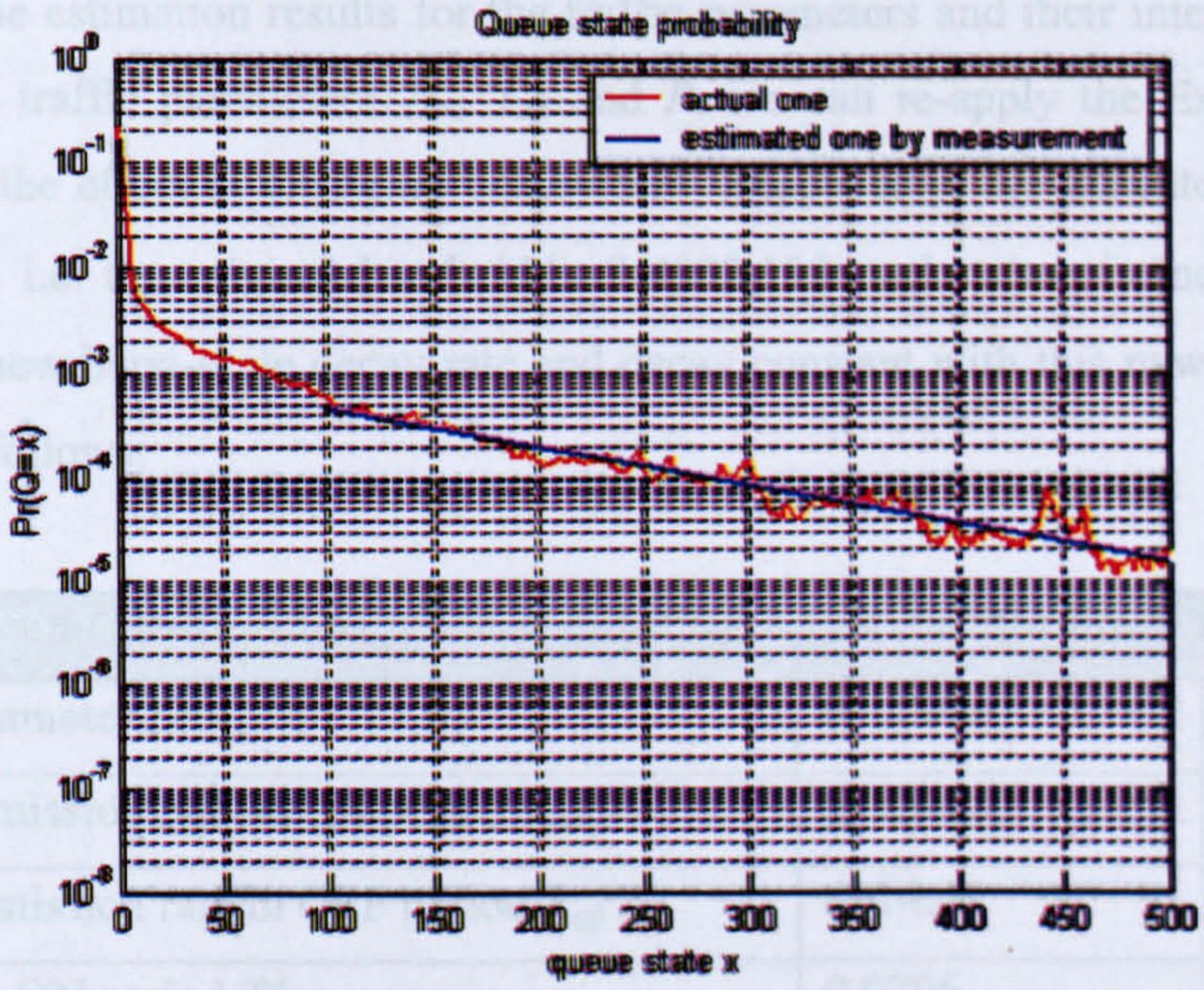


Figure 8-8 Comparison between true and measured burst-scale decay rate and constant

Estimated parameter by measurement	Estimated value	Formula
1. Load ρ	0.843	8.16
2. packet-scale decay rate η_p	0.7242	8.1
3. activity factor α	0.355	$(C\rho)/(NR)$
4. packet-scale & burst-scale interception point x_{pb}	16	8.17
5. transmission rate R	169.6	8.15
6. intermediate parameter A	24.852	Table 8.1 g
7. intermediate parameter A_p	4200	Table 8.1 e
8. intermediate parameter D	0.3249	Table 8.1 i
9. Aggregate transmission rate in ON period R_{on}	5889.6	Table 8.1 c
10 Aggregate transmission rate in OFF period R_{off}	3386.8	Table 8.1 d
11 parameter H	907.4331	$R_{on} - C$
12 parameter G	3314.8	$(1\text{-}D)(C\text{-}R_{off})/D$
13 Aggregate ON time $T_{(on)}$	0.07	8.20
14 Mean ON time T_{on}	0.324	$\rho C/(NR)$
15 Aggregate OFF time $T_{(off)}$	0.145	Table 8.1b
16 Mean OFF time T_{off}	0.588	$(1/\alpha\text{-}1)T_{on}$

Table 8-2 The intermediate parameters and traffic parameters estimation

Table 8-2 shows the estimation results for the traffic parameters and their intermediate parameters. After knowing the traffic parameters T_{on}, T_{off} and R, we can re-apply the Excess Rate analytical model to estimate the effect of changing bandwidth. Supposedly, the allocated bandwidth will be increased by 21%, i.e. the original bandwidth C 4982.195 packets/sec is increased to C' 6049.8 packets/sec. The new burst-scale decay rate and decay constant with this new allocated bandwidth are determined as follows:

Estimated parameter by measurement	Estimated value	Formula
1. intermediate parameter D'	0.0.00367	Table 8.1 i
2. Aggregate transmission rate in ON period R_{on}'	6433.4	Table 8.1 c
3. Aggregate transmission rate in OFF period R_{off}'	4114.4	Table 8.1 d
4. Aggregate mean ON period $T'_{(on)}$	0.0296	Table 8.1 a
5. Aggregate mean OFF period $T'_{(off)}$	0.7758	Table 8.1 b
6. burst-scale decay rate η_b'	0.964	8.10
7. burst-scale decay constant c_b'	$2.1592 \cdot 10^{-4}$	8.22

Table 8-3 The intermediate parameters, burst-scale decay rate and constant calculation with respect to the new adjusted service rate C'

The above table shows how to calculate the burst-scale decay rate and decay constant with respect to new adjusted bandwidth C'. It can be noted that the decay rate decreases from 0.9917 to 0.964 after increasing the bandwidth. We perform a simulation with the new bandwidth to compare the new burst-scale decay rate and decay constant. The comparison result is shown as in Figure 8-9. With reference to Figure 8-9, the blue line represents the true queue length distribution with the new bandwidth, whereas the red line represents our estimate obtained from our analytical model with the measurement data. It can be seen that the estimated queue length distribution closely matches with the true one. It illustrates that our measurement-analytical scheme is quite promising to estimate the effect of changing bandwidth on QoS performance.

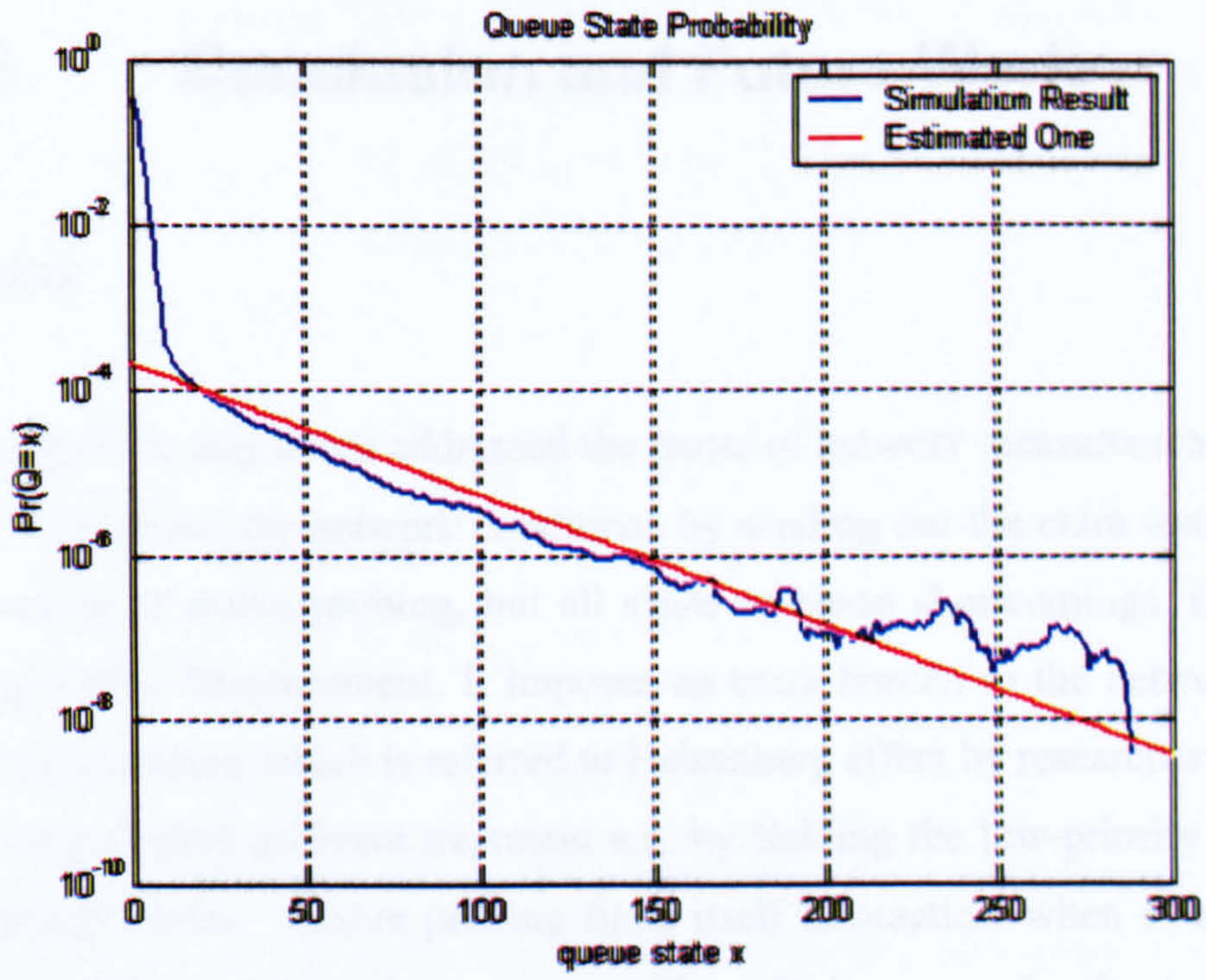


Figure 8-9 Comparison the true queue length distribution and the estimated one after changing Bandwidth

8.6 Summary

In this chapter, we illustrated an application of the measurement (non-intrusive type) on the bandwidth adjustment issue. This technique combines the QL Measurement method developed in Chapter 4 and the Excess-Rate analytical model. We first improved the accuracy of the analytical model, introducing the factor of the packet-scale and the burst-scale interception point in the original formula with explanation. Simulation results show that the accuracy of the analytical model in capturing the decay rate can be greatly improved after introducing this factor, hence the improvement in our proposed analytical-measurement scheme. The main idea of this analytical-measurement scheme is to capture the required traffic characteristics by measurement, and then reapply this set of traffic parameters for the calculation with respect to the new allocated bandwidth. From the simulation results, this shows that our scheme is quite promising in estimating the new queue length distribution for changed bandwidth.

Chapter 9 Conclusion and Future Work

9.1 Conclusion

The research presented in this thesis addressed the issues of network measurement. Active probing is a prevalent tool to examine the network conditions by sending out the extra testing packets. There are many derivatives of active probing, but all share common shortcomings. Problems may exist when employing Active Measurement. It imposes an extra burden to the network and distorts the original measuring condition which is referred to Heisenberg effect by researchers. Furthermore, the testing packets may receive different treatment e.g. by fielding the low-priority tag. This results in wrong measurement results. Active probing finds itself impractical when every customer's SLA validation is indeed carried out, then much traffic will be created. Because of this, Passive Measurement attracts much attention for the network measurement, and this has motivated our research work.

In our scheme, the network element is passively monitored in Passive Measurement and so it does not impose any intrusive effect. The objective of our research was the full formulation of a new Passive Measurement scheme: QL Measurement. In our measurement framework, the QL Measurement scheme is implemented in each node. Our approach exploits the known characteristics of fundamental queue length distributions of a FIFO queue multiplexing Markovian and Power-law traffic respectively. These are also applied to WRR and WFQ queuing discipline. The goal is to capture the per-hop queue length distribution by using a non-intrusive measurement approach. Literature reviews reveal that the queue length distributions can be well described by two models in many traffic scenarios. Thanks to these queue length distribution models: exponential tail model and power-law tail model, they allow us to develop the QL scheme with low computational complexity, low-overhead and little data storage requirement, while retaining a good accuracy. The simplicity of QL Measurement makes it suitable to be adopted in real practice.

Accurate estimation of the per-hop queue length distribution is crucial. Literature reviews suggest the importance of the per-hop queue length distribution. Firstly, the per-hop queue length distribution correlates the packet delay in the single hop. The per-hop queue length distribution can serve to identify the problem in the network and find out the reason behind QoS degradation. In addition, the per-hop queue length distributions can be used to deduce the end-to-end delay distribution. Experimental results reveal the links' delay independence in many networks [VLE95]

[SEU00]. This suggests that convolving successive delay distributions predicts the total end-to-end delay distribution with considerable accuracy. In other words, by convolving queue length distributions, the counterpart of the delay distributions, along the path is also a solution. The knowledge of the end-to-end delay distribution is useful for the network performance assessment like SLA's latency validation.

As described in the body of this thesis, QL Measurement comprises only addition and comparison operations. The queue length is passively monitored and compared with a pre-defined partition point for every incoming packet. According to the comparison result, the relating measurement data can be updated.

This thesis evaluates QL Measurement in detail. How the partition point is selected is a fundamental issue in adopting the QL Measurement scheme. The purpose of this partition point is to isolate the packet-scale region from the burst-scale region during measurement. The impact of the partition point on the per-hop queue length estimation was investigated through a number of simulations. We compared the actual queue length distribution with the estimate one obtained by the QL method. The simulation results suggest that the estimation result is insensitive to the partition point provided that it is large enough to be within the burst-scale. Since the packet-scale region is normally within the range of the order of a few 10's of packets, in the thesis, we argue that the reasonable value of the partition point is larger than 20 but smaller than 100. As there is no stringent constraint on partition point selection, it provides the network operator with more freedom to make the selection and provides for ease of implementation in practice.

.

We compared QL Measurement with other sophisticated techniques, such as LDT and MVT. Unlike our QL Measurement scheme, the approach in these takes the statistics of the packet arrival's volume instead of direct queue length monitoring. But similar to our approach, both techniques provide a solution for the queue length distribution from measurement results. In these techniques, the data storage for the measurement result is demanding and the computational complexity is high. This is unfavourable for them to be deployed in real networks. By visual assessment the estimate queue length distribution by the QL scheme with LDT or MVT, the results show that the QL Measurement scheme gives better estimation result compared with these schemes. Large Deviation Theory has difficulties in determining the decay constant, and so this leads to a big discrepancy from the actual queue length distribution. For MVT, a discrepancy also exists between the actual and the estimate queue length distribution. The discrepancy gets worse when approaching the tail of the queue length distribution.

To estimate the end-to-end delay distribution, it is possible to perform convolution of every per-hop queue length distribution along the path when the same bandwidth is received at each queue. Therefore, we formulated the normalization process before the convolution process when the per-hop service rates along the path are not the same. The idea of the normalization process is to adjust the estimated queue length distribution to reflect the real delay time with respect to a reference link bandwidth instead. Simulation results proved that this approach is quite promising. However, in a link bandwidth sharing system, like Class-based queuing or DiffServ, the link bandwidth is divided and allocated to each queue by a scheduler. It is not easy to know the actual service rate received by a queue. The actual service rate received is a function of the traffic condition on the other queues. Although the minimum service rate is guaranteed as the multiple of the weight with the link service rate, there may exist a big difference between the minimum guaranteed bandwidth and the actual received bandwidth. Therefore, we introduced BW Measurement for estimating the actual received bandwidth. The measurement result will be used in conjunction with QL Measurement for the end-to-end delay estimation. We examined the effectiveness of QL+BW Measurement scheme under various conditions: different scheduler – Weighted Round Robin (WRR) and Weighted Fair Queuing (WFQ), different scheduler weighting, different load conditions and different hop counts. By visual assessment between the actual end-to-end delay distributions with the estimates from QL+BW Measurement scheme, results show that this approach is capable of providing an estimate with good accuracy.

We compared the effectiveness of QL Measurement and active probing in capturing the tail of the end-to-end delay distribution. The results show that active probing requires substantial amount of testing packets in order to capture the tail of the delay distribution, but QL Measurement produces more promising results. We also applied QL estimation to the SLA's end-to-end latency verification and compared this with active probing. To make the comparison, we borrowed the idea outlined in the document of ATM Traffic Management. With respect to a specific delay target, the tail of the delay distribution indicates the portion of the packets violating this delay target. This portion is referred to as α-tail. In other words, the value α indicates the percentage of the packets violating the required target value. Certainly, the more stringent the target is, the smaller α will be. We measured the estimation error produced by QL Measurement and active probing with a specific delay target. According to the simulation results, it suggests that the QL Measurement scheme provides the result with the estimation error within the order of magnitude with α and reveal that QL Measurement provides better estimation results than active probing in most circumstances. In addition, when the delay target is stringent or the percentage of violating packets is very small, then active probing is unlikely to provide the estimation result as it demands a large amount of probing packet to test. But, this inevitably creates much extra load to the network.

Apart from the end-to-end delay distribution measurement, the application of QL Measurement was further explored. Literature reviews address the importance of bandwidth adjustment. This ensures that the network element operates in an optimum way. Based on QL Measurement, we proposed a technique to determine the effect of changing bandwidth. Our approach combines the analytical model and the measurement methods. Excess-Rate theory provides a close-form analytical formula for the decay rate estimation. This formula is then further enhanced by taking the effect of packet-scale queuing behaviour into consideration. The accuracy of enhanced formula was assessed and presented in the thesis. Based on this new formula, the traffic parameters are estimated with the burst-scale decay rate and constant measured by QL Measurement together with the load measurement and traffic volume measurement. The new queue length distribution with respect to the new bandwidth is predicted by re-applying the new formula with the traffic parameters. Then, the effect of changing bandwidth can be predicted. In this thesis, we explained how to perform this technique and the accuracy was assessed.

9.2 Future Work

We introduced a novel Passive Measurement method QL Measurement for per-hop queue length distribution. The queue is passively monitored at the local node. The merits of QL Measurement are its simplicity, small overhead and low measurement data requirements. Hence, the fast packet forwarding at the node will not be interfered with when adopting our QL Measurement scheme. In this thesis, we fully explained how to perform QL Measurement to estimate the per-hop queue length distribution and the end-to-end delay distribution. Additionally, the further application of QL Measurement in bandwidth adjustment was also introduced. In this thesis, the features or properties of QL Measurement were investigated. The simulation results show that the QL scheme is very promising. Certainly, there are future works in different directions that can benefit this research work.

We compared our scheme with count-based Active probing. We focus on the effectiveness of capturing the end-to-end delay distribution and the SLA latency verification. Further work is beneficial by performing a more comprehensive comparison in various aspects such as fairness, overhead, complexity to see if passive measurement is still superior to active probing.

In our measurement QL and BW, the measurement results represent the average value of our measurement targets. We have not considered the second moment of the statistic which may be useful as well. Further investigation is necessary.

In this research work, we exploit the queuing characteristics of a FIFO queue multiplexing either Markovian or Power-law traffic. The queue tail can be modelled as exponential decay or power-law decay model. Although FIFO queuing discipline is widely employed, some active queue management schemes like Random Early Detection (RED) are also possible. Simulation results suggest that the queue length distribution for a RED queue still has the packet-scale queuing region and the burst-scale queuing region, but the tail drops rapidly and it is in a shape of a "waterfall" (see Appendix-E). In other words, the queue length distribution is divided into three regions: packet-scale queuing region, burst-scale queuing region and an additional region at the tail. A new model is necessary to approximate this tail region.

In this thesis, we assume the packet loss is small by assuming a large buffer. If the packet loss is significant, the queue tail will be modified substantially. Appendix-F shows the queue length distribution with significant packet loss. Again, the queue length distribution still has the packet-scale queuing region, burst-scale queuing region and the queue tail. There is a sudden spike at the tail. If the same rationale for the case of RED queue applies, we then suggest that the queue length distribution should be divided into three regions and an additional measurement is necessary to capture the tail.

In this research work, we used the queue length distribution models for the Markovian traffic (SRD) and the Power-law traffic (LRD), and we have different analytical formulas handling the measurement results from QL Measurement. When the traffic characteristics are not known in advance, some additional measurement is necessary to identify the traffic type and so the appropriate analytical formula is selected. Hurst parameter is an effective index to indicate the burstiness of the traffic, or to differentiate SRD and LRD traffic. A novel technique of on-line Hurst parameter measurement was proposed in the work [ROU98]. It is possible to integrate this measurement scheme into our QL scheme. The Hurst parameter measurement result is stored together with QL measurement results. Then, the Hurst parameter measurement tells which queue length distribution model should be used. This approach is interesting but it certainly requires further work to study the feasibility. However, when the measurement data for the traffic characteristics like Hurst parameter is not available, then we may not have the knowledge of how the queue state probability behaves. Under these circumstances, we may need to make a prior assumption of the model for the queue

state distribution. It is interesting to investigate the performance and the accuracy of using Exponential decay queue tail model for Power-law traffic and vice versa.

Sensitivity analysis is a procedure to determine the sensitivity of the outcomes of an alternative to changes in its parameter. If a small change in a parameter results in relatively large changes in the outcomes, then the outcomes are said to be sensitive to that parameter. In this work, we studied the sensitivity analysis of QL and BW measurement with respect to their parameters like partition point, selection criterion, the queue's utilization condition, traffic sources' parameters like shape parameter, number of sources etc. Further sensitivity analysis should be performed to study the effectiveness of our measurement scheme like various setting for ON/OFF time, the transmission rate of the source models.

In the DiffServ mechanism, computational complexity is pushed to the edge node to ensure the fast forwarding operation in the core network. Analogous to this approach, in our scheme, the burden is laid on the centralised network management centre or network operation centre (NOC), and so it does not obstruct the forwarding operation in the switch/router when employing QL Measurement. The workload of the convolving process can be quite heavy. Although the primary goal of our scheme is not aiming at on-line end-to-end delay distribution estimation, the convolving process can be off-line performed in the NOC. Certainly, it is always advantageous if the workload of the NOC can be reduced. A novel fast convolution technique was proposed in [SEU00]. The complexity of the convolving process can be greatly reduced when convolving the per-hop delay distribution at the expense of accuracy. Therefore, it is also interesting to integrate this scheme with ours. Further work should be performed to see how much workload can be reduced and how much accuracy can be retained.

APPENDICES

Appendix A. Definition of ICMP Header

Type	Name	Code(s)
0		
1-2	Unassigned	
3	Destination unreachable	0-Net unreachable
3		1-Host unreachable
		2-Protocol unreachable
		3-Port unreachable
		4-Fragmentation needed and DF bit set
		5-Source route failed
		6-Destination network unknown
		7-Destination host unknown
		8-Source host isolated
		9-Communication with destination network is administratively
		10-Communication with destination host is administratively
		11-Destination network unreachable for TOS
		12-Destination host unreachable for TOS
4	Source quench	0-none
5	Redirect	0-Redirect datagram for the network
		1-Redirect datagram for the host
		2-Redirect datagram for the TOS and network
6	Alternate host address	0-Alternate address for host
7	Unassigned	
8	Echo request	0-None
9	Router advertisment	0-None
10	Router selection	0-None
11	Time Exceeded	0-Time to live exceeded in transit
		1-Fragment reassembly time exceeded
12	Parameter problem	0-Pointer indicates the error
		1-Missing a required optioin
		2-Bad length
13	Timestamp	0-None
14	Timestamp reply	0-None
15	Information request	0-None
16	Information reply	0-None
17	Address mask request	0-None
18	Address mask reply	0-None
19	Reserved (for security)	
20-29	Reserved (for experiment)	
30	Traceroute	
31	Datagram conversion error	
32	Mobile host redirect	
33	IPv6 where-are-you	
34	IPv6 I-am-here	
35	Mobile registration request	

Appendix B. Proof of Equation (4-15)

As shown in Equation (4-15), $\bar{q}_{high}$ is given as follows:

$$\bar{q}_{high} = \sum_{j=q_p+1}^{\infty} j \frac{c_{bm}\eta_{bm}{}^{j}}{\sum_{k=q_p+1}^{\infty} c_{bm}\eta_{bm}{}^{k}}$$

$$= \frac{\sum_{j=q_p+1}^{\infty} j \cdot \eta_{bm}{}^{j}}{\sum_{k=q_p+1}^{\infty} \eta_{bm}{}^{k}}$$

$$= \frac{\eta_{bm} \frac{d}{d\eta_{bm}} \sum_{j=q_p+1}^{\infty} \eta_{bm}{}^{j}}{\sum_{k=q_p+1}^{\infty} \eta_{bm}{}^{k}}$$

$$= \frac{\eta_{bm} \frac{d}{d\eta_{bm}} \left(\frac{1}{1-\eta_{bm}} - M \right)}{\frac{1}{1-\eta_{bm}} - M} \qquad \text{(B-1)}$$

where $M = 1 + \eta_{bm} + \eta_{bm}{}^{2} + \ldots + \eta_{bm}{}^{q_p} = \frac{1-\eta_{bm}{}^{q_p+1}}{1-\eta_{bm}}$

Substitute M into Equation (B-1)

$$\bar{q}_{high} = \frac{\eta_{bm} \left(\frac{1}{(1-\eta_{bm})^2} - \frac{dM}{d\eta_{bm}} \right)}{\frac{\eta_{bm}{}^{q_p+1}}{1-\eta_{bm}}} \qquad \text{(B-2)}$$

Since $\frac{dM}{d\eta_{bm}} = \frac{-(q_p+1)\cdot \eta_{bm}{}^{q_p}}{1-\eta_{bm}} + \frac{1-\eta_{bm}{}^{q_p+1}}{(1-\eta_{bm})^2} = \frac{1-(1-\eta_{bm})(q_p+1)\eta_{bm}{}^{q_p} - \eta_{bm}{}^{q_p+1}}{(1-\eta_{bm})^2}$ (B-3)

Substitute back equation (B-2)

$$\bar{q}_{high} = \frac{(1-\eta_{bm})(q_p+1)+\eta_{bm}}{1-\eta_{bm}}$$

$$= q_p + \frac{1}{1-\eta_{bm}}$$

Therefore,

$$\eta_{bm} = 1 - \frac{1}{q_{high} - q_p} \qquad \text{(B-4)}$$

Appendix C. Maximum Likelihood Estimation of Burst-scale Decay Rate

Let $x_1, x_2, \ldots, x_n$ are n samples of the queue length seen by the packet arrival and the queue length is greater than the partition point.

According to the queue tail distribution model in Equation 4-6,

$$Q(x) = c_{bm}\eta_{bm}{}^{x} \tag{C-1}$$

The conditional probability of queue tail probability provided that it is greater than the partition point is given as by assuming the queue length is continuous:

$$Q(x \mid x > q_p) = \frac{c_{bm}\eta_{bm}{}^{x}}{\int_{q_p+1}^{\infty} c_{bm}\eta_{bm}{}^{x} dx} \tag{C-2}$$

Let's define a new variable θ such that $\eta_{bm} = e^{-\theta}$ and substitute into Equation (C-2)

$$Q(x \mid x > q_p) = \frac{c_{bm}e^{-\theta x}}{\int_{q_p+1}^{\infty} c_{bm}e^{-\theta x} dx} = \theta e^{-\theta\left(x-(q_p+1)\right)} \tag{C-3}$$

Then, the likelihood function $L(\theta)$ of the n measurement samples is defined as:

$$L(\theta) = \prod_{i=1}^{n} \theta e^{-\theta\left(x_i-(q_p+1)\right)} \tag{C-4}$$

The maximum-likelihood estimator of θ is given as $\theta_{ML} = \max_{\theta} \ln L(\theta)$. This can be achieved by finding the solution for the differentiation equation of (C-4).

$$\left.\frac{\partial \ln L(\theta)}{\partial \theta}\right|_{\theta=\theta_{ML}} = 0 \tag{C-5}$$

$$\frac{\partial}{\partial \theta}\left(n\ln\theta - \theta\left(\sum_{i=1}^{n} x_i - n\left(q_p+1\right)\right)\right) = 0$$

$$\frac{n}{\theta} - \left(\sum_{i=1}^{n} x_i - n\left(q_p+1\right)\right) = 0$$

Therefore, θ_{ML} is obtained as follows:

$$\theta_{ML} = \frac{1}{\frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) - \left(q_p + 1\right)} \tag{C-6}$$

Noted that $\sum_{i=1}^{n} x_i$ is equivalent to our measurement result $queue_{high}$, whereas n is same as the measurement result $freq_{high}$. As a result, the factor $\frac{1}{n}\sum_{i=1}^{n} x_i$ in the denominator of Equation (C-6) is same as $\bar{q}_{high}$ as in Equation (4-13). Since $\eta_{bm} = e^{-\theta}$, therefore, the maximum likelihood estimator of η_{bm} is given as:

$$\eta_{bm} = e^{-\frac{1}{\bar{q}_{high} - \left(q_p + 1\right)}} \tag{C-7}$$

The expression is expanded by Talyor's series

$$\eta_{bm} = 1 - \frac{1}{\bar{q}_{high} - \left(q_p + 1\right)} + \frac{1}{2!}\left(\frac{1}{\bar{q}_{high} - \left(q_p + 1\right)}\right)^2 - \ldots \tag{C-8}$$

Taking the first order in the series and q_p is greater than 1, then Equation C-8 becomes:

$$\eta_{bm} \approx 1 - \frac{1}{\bar{q}_{high} - q_p} \tag{C-9}$$

As a result, the maximum likelihood estimator is same as Equation (4-15)

Appendix D. Variable Packet Size with Bimodal Distribution Simulation

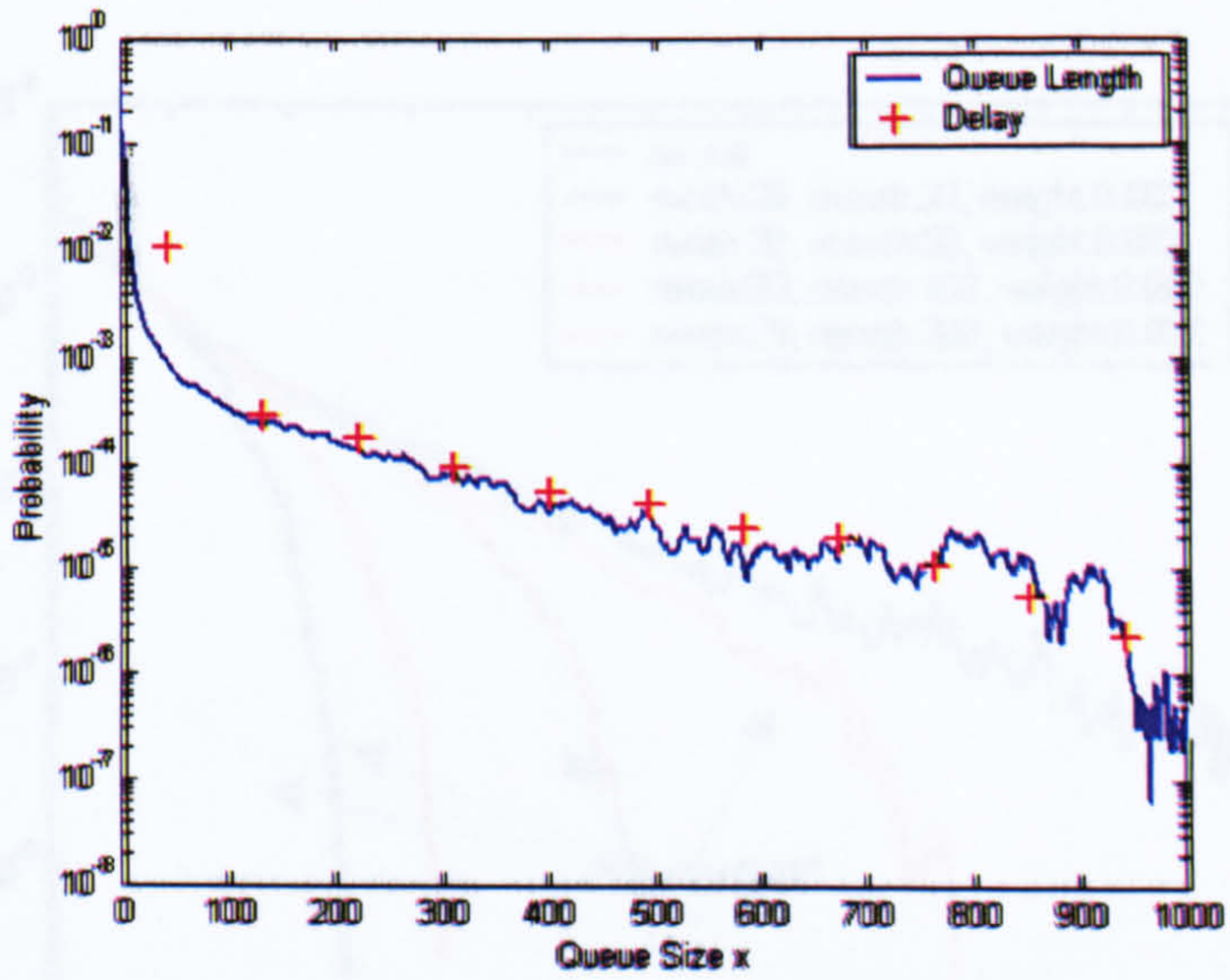


Figure D-1 Comparison between delay and queue length distribution (Markovian Traffic)

The experimental setup for Figure D-1 is as follows:

70 ON/OFF sources, Mean On time: 0.96 sec (exponential), Mean Off time: 1.69 sec (exponential), Mean packet size: 770-byte-long (bimodal distribution: 40 and 1500 byte), Transmission rate: 32Mbps. The packet delay is converted to the queue size by using mean packet size.

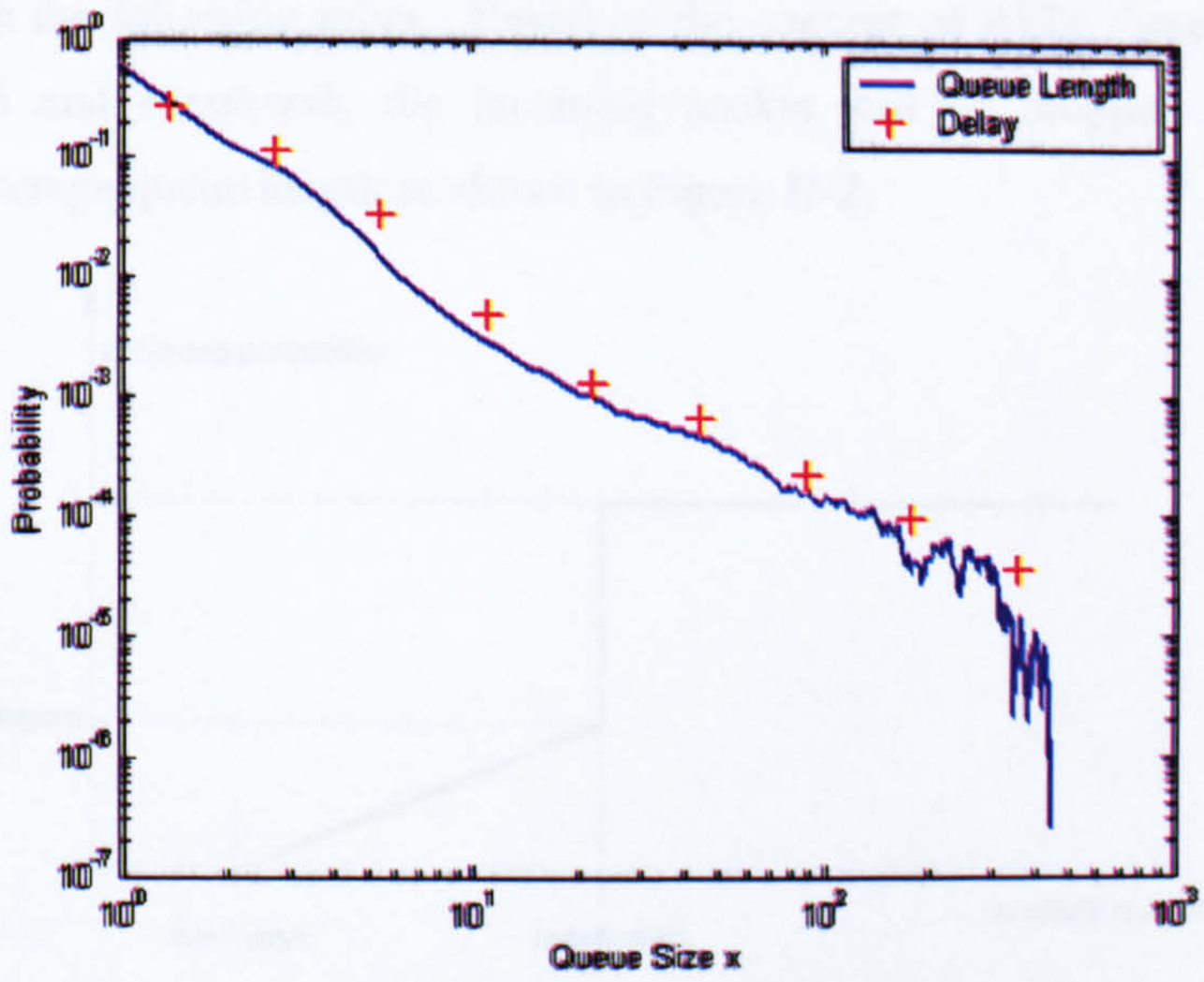


Figure D-2 Comparison between delay and queue length distribution (Power-law Traffic)

The experimental setup for Figure D-1 is as follows:

100 ON/OFF sources, Mean On time: 1 sec (Pareto), Mean Off time: 10 sec (Pareto), Shape parameter: 1.4, Mean packet size: 770-byte-long (bimodal distribution: 40 and 1500 byte), Transmission rate: 11.6Mbps. The packet delay is converted to queue size by using mean packet size.

Appendix E. An Example of Queue Length Distribution of a buffer with RED

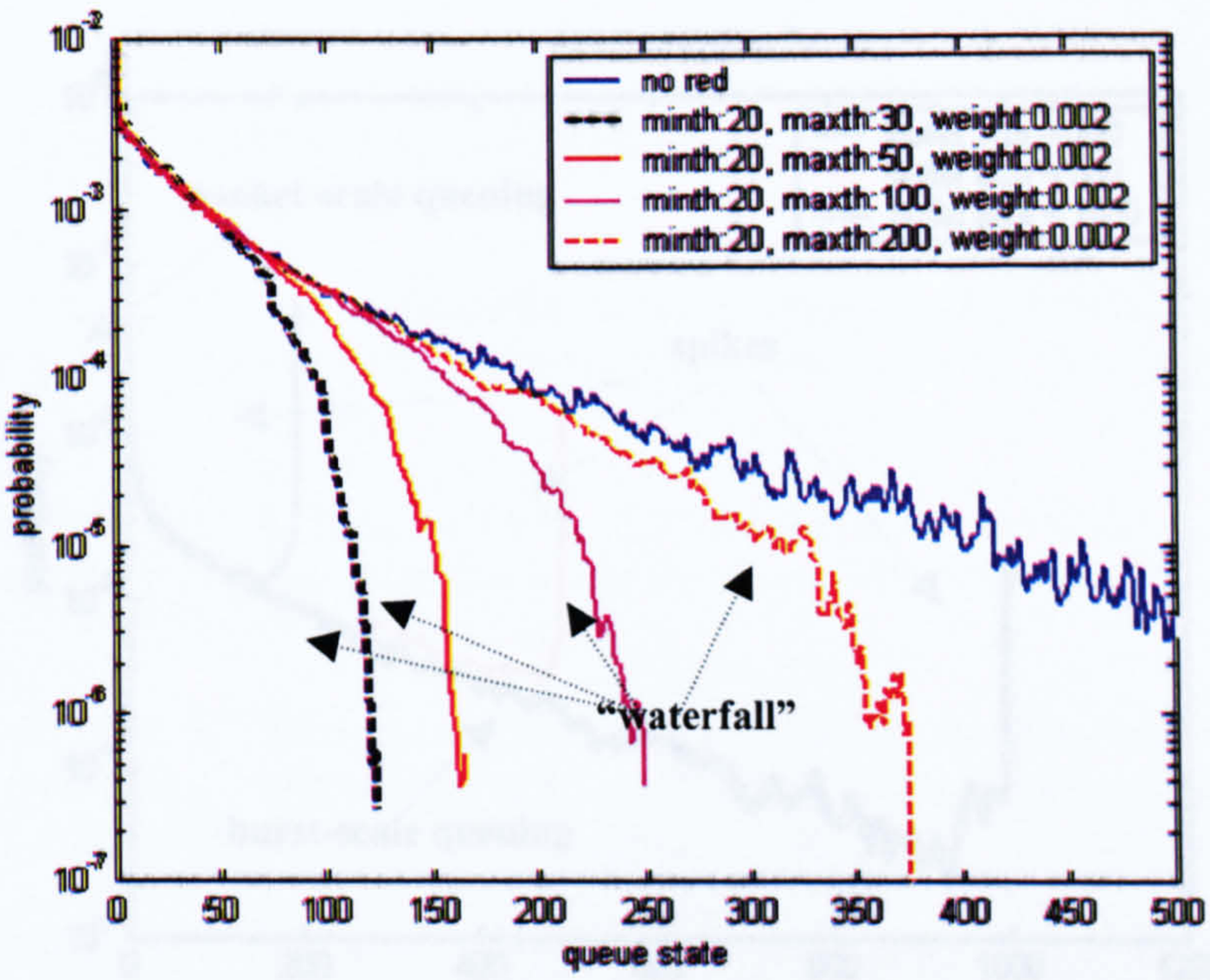


Figure E-1 A typical example of queue length distribution for a RED queue

Figure E-1 shows a typical example of the queue length distribution of a RED queue. The results were obtained with the following setup Based on the concept of RED, there exists two threshold such as *minthresh* and *maxthresh*, the incoming packet will be dropped with the probability according to the average queue length as shown in Figure D-2.

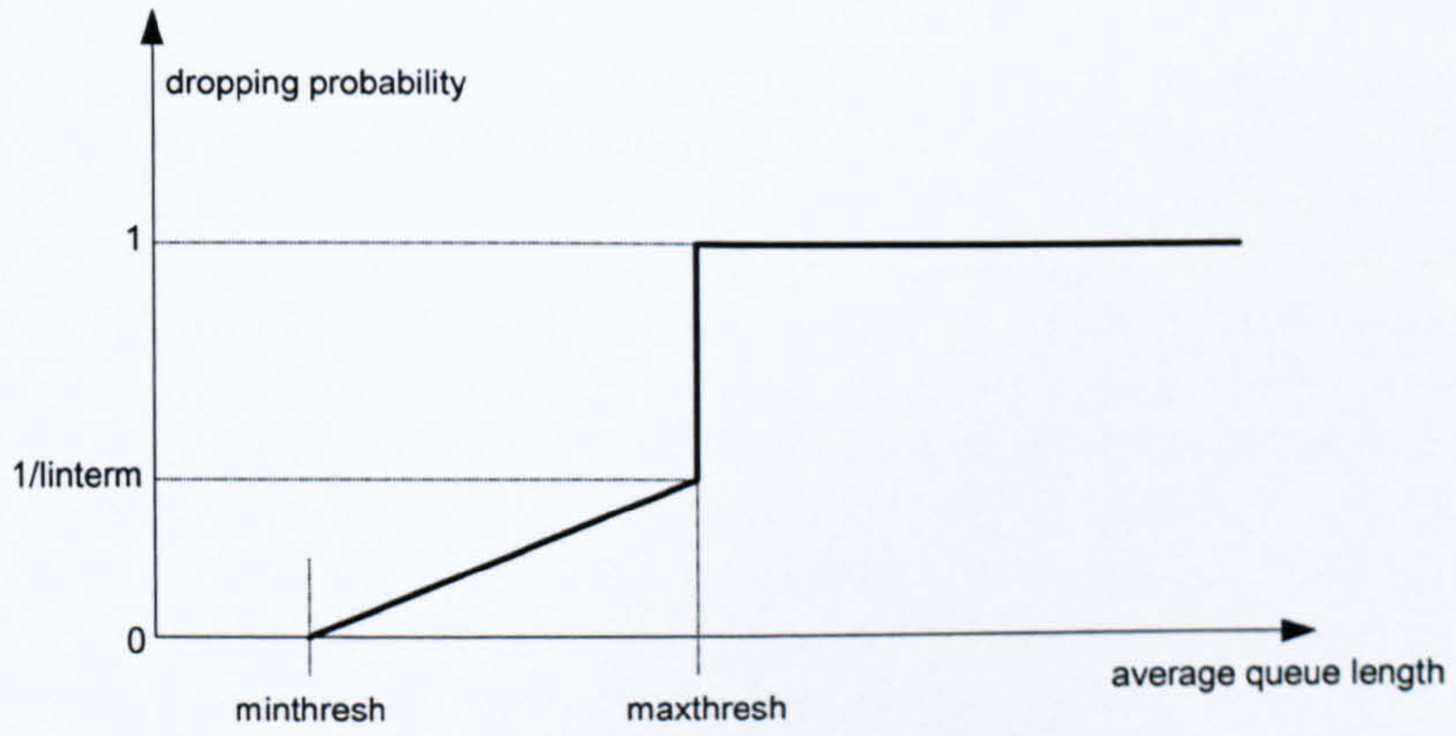


Figure E-2 The dropping probability of RED

The RED queue is multiplexing 10 Markovian ON/OFF sources. The parameter *linterm* = 10 for the RED. Figure D-1 shows the results for different value of *minthresh* and *maxthresh*.

Appendix F. An Example of Queue Length Distribution with substantial packet loss

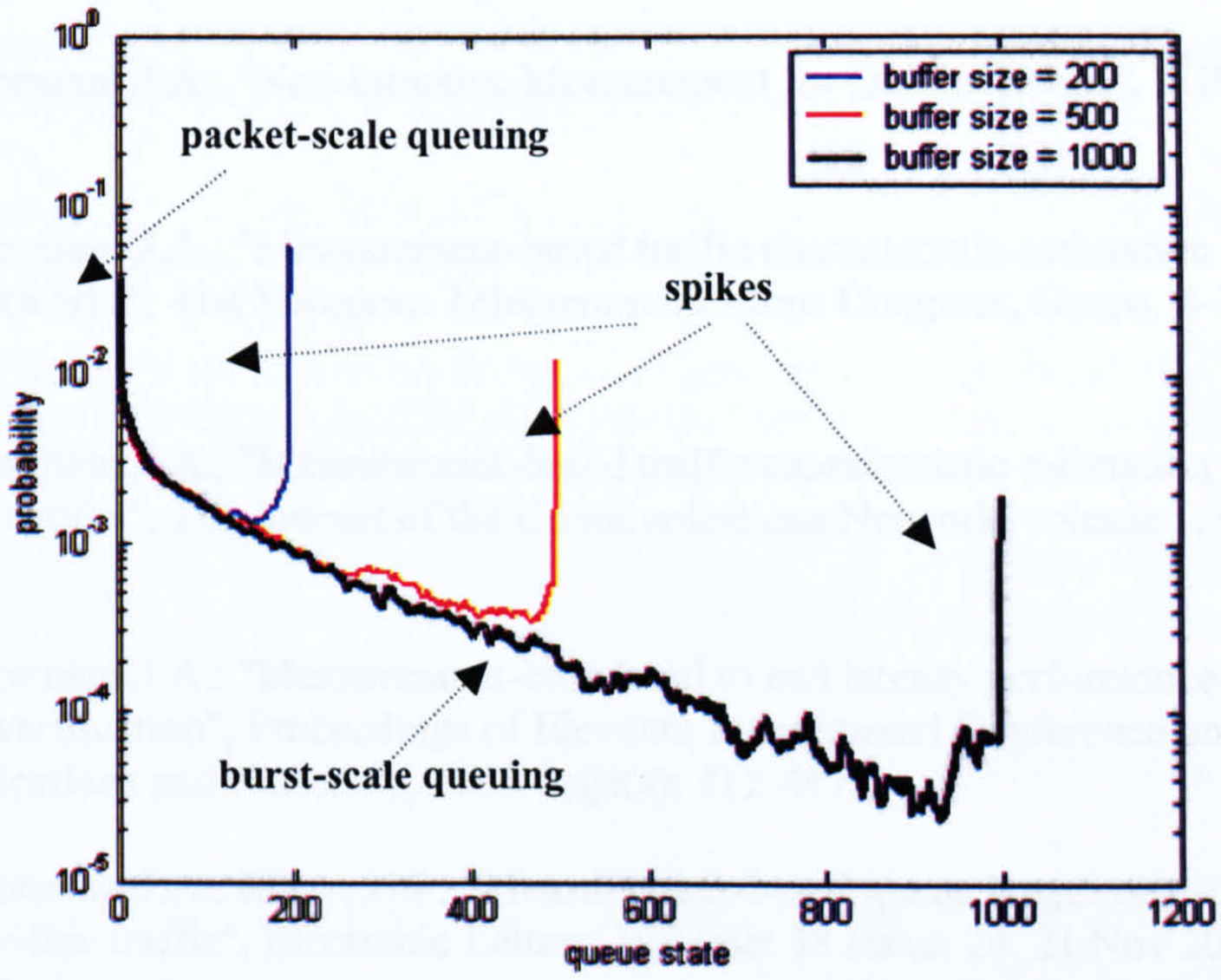


Figure F-1 An example of the queue length distribution with substantial packet loss

The experimental-setup is shown as follows:

10 ON/OFF sources, Mean On time: 0.96 sec (exponential), Mean Off time: 1.69 sec (exponential), packet size: 53 bytes, transmission rate during On period: 70808 bit/sec, load: 0.8.

Authors Publications

1. Leung,C.M.; Schormans,J.A.; "Non-intrusive Measurement for QoS inference", PGNet 2002 page 24-39.

2. Leung,C.M.; Schormans,J.A.; "Measurement-based traffic characteristic estimation for QoS Oriented IP Networks", 41st European Telecommunications Congress, Genoa, 4-7 September 2002.

3. Leung,C.M.; Schormans,J.A.; "Measurement-based traffic characteristic estimation for QoS Oriented IP Networks", The Journal of the Communications Network, volume 1, part2, July-Sept 2002.

4. Leung,C.M.; Schormans,J.A.; "Measurement-based end to end latency performance prediction for SLA verification", Proceedings of Eleventh International Conference on Computer Communications and Networks, 2002 Page(s): 412-417

5. Leung,C.M.; Schormans,J.A.; Ma, A.H.I.; "Measurement-based queue length distribution estimation for power-law traffic", Electronic Letters, Volume: 38 Issue: 24, 21 Nov 2002 Pages(s): 1608-1610

6. Leung,C.M.; Schormans,J.A.; Pitts,J.M.; Woolf,A.; "Accurate decay rate prediction for burst-scale queueing in packet buffers", Electronic Letter, Volume: 39 Issue:2, 23 Jan 2003, Page(s):253-254

7. Timotijevic,T; Leung,C.M.; Schormans,J.A.; "Accuracy of measrement techniques supporting QoS in packet based Intranet and Extranet VPNs" accepted by IEE Proc. Communications Special Issue Superhighways Technology and Broadband VPN Volume 150, Issue 4.

8. Leung,C.M.; Schormans,J.A..; "Measurement-based end to end latency performance prediction for SLA verification", accepted by Journal of System and software

References

[BAR02] Barford, Paul; Sommers, Joel.; *A Comparison of Active and Passive Methods for Measuring Packet Loss*, UW Technical Report, October, 2002.

[ØST02] ØsterbØ, O.; *An approximate method to calculate the distribution of end-to-end delay in packet networks*, Telenor scientific report, 2002

[ABA95] Abate, J.: Choudhury, G.L.; Whitt, W.; *Exponential Approximations for Tail Probabilities in Queues, I: Waiting Times*, Operations Research, vol. 43, pp. 885-901, 1995

Abate, J.; Choudhury, G.L.; Whitt, W.; *Exponential Approximations for Tail Probabilities in Queues, II: Sojourn Time and Workload*, Operations Research, vol. 44, pp. 758-763, 1996

[ADA97] Adas, A.; *Traffic models in broadband networks*, IEEE Communications, vol. 35(7), pp. 82-89, 1997

[ADV99] Advanced Org.; *Participants guide to the Surveyor daily summary reports*, available from

http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-1999/cucs-015-99.ps.gz

[AGI00] Agilent technology; *Packet over Sonet/SDH (POS) functional testing*, product note, 2000

[AID02] Aida, M.; Ishibashi, K.; Kanazawa, T.; *Compact-monitor: Change-of-measure based passive/active monitoring weigted active sampling scheme to infer QoS*, Proc. of the 2002 Symposium on Applications and the Internet Workshops, 2002

[ATM96] The ATM Forum; *Traffic Management Specification Version 4.0*, af-tm-0056.000, Apr. 1996

[BAI91] Baiocchi, A.; Melazzi, N.; Listanti, M., Roveri, A.; Winkler, R.; *Loss performance analysis of an ATM multiplexer loaded with high-speed ON-OFF sources*, IEEE Journal on Selected Areas in Communications, vol. 9(3), pp. 388-393, Apr. 1991

[BHA00] Bhadra, A.; Sadiku, M.N.O.; *Simulation of an ATM network using on-off model*, Proc. of the IEEE, pp. 467-470, Apr. 2000

[BLA98] Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W.; *An Architecture for Differentiated Services*, RFC2475, Internet Request For Comments, IETF, Dec. 1998

[BOV02] Bovy, C.J.; Mertodimedjo, H.T.; Hooghiemstra, G.; Uijterwaal, H., Van Mieghem, P.; *Analysis of End-to-end Delay Measurements in Internet*, Passive and Active Measurement Workshop, PAM2002

[BRE00] Breslau, L.; Estrin, D.; Fall, K.; Floyd, S., Heidemann, J.; Helmy, A.; Huang, P.; McCanne, S.; Varadhan, K.; Xu, Y.; Yu, H.; *Advances in Network Simulation*, IEEE Computer, May 2000

[BRE00a] Breslau, L.; Knightly, E.; Shenker, S.; Stoica, I.; Zhang, H.; *Endpoint Admission Control: Architectural Issues and Performance*, Proc. ACM SIGCOMM 2000, Aug. 2000

[BRO01] Brownlee, Nevil; Lossley Chris.; *Fundamentals of Internet Measurement: A Tutorial*, Keynote Systems, 2001.

[CAC99] Caceres, R.; Duffield, N.G.; Horowitz, J.; Towsley, D.; Bu, T.; *Multicast-Based Inference of Network Internal Characteristics: Accuracy of Packet Loss Estimation*, IEEE INFOCOMM'99, Mar. 1999

[CHA00] Chan, M.C.; Lin, Y.J.; Wang, X.; *A scalable monitoring approach for service level agreements validation*, International Conference on Network Protocols, pp. 37-48, Nov., 2000

[CHAN00] Chang C.S.; *Performance guarantees in communication networks*, Springer-Verlag, 2000

[CHO97] Choe, J.; Shroff, N.B.; *A New Method to Determine the Queue Length Distribution at an ATM Multiplexer*, IEEE INFOCOM '97, pp. 549-556, 1997

[CIS03] Cisco – *Measuring Delay, Jitter, and Packet Loss with Cisco IOS SAA and RTTON* available from

http://www.cisco.com/warp/public/126/saa.pdf

[CIS03a] Cisco – *Service-level Management: Defining and Monitoring Service levels in the Enterprise* available from

http://www.cisco.com/warp/public/cc/pd/wr2k/svmnso/prodlit/srlm_wp.pdf

[CLA93] Claffy, K.C.; Polyzos, G.C.; Braun, H.W.; *Measurement consideration for assessing unidirectional latencies*, Technical Report UCSD Report CS92-252, SDSC Report GA-A21018, Computer Systems Laboratory, UCSD, 1993

[CLA93a] Claffy, K.C.; Polyzos, G.C.; Braun, H.W.; *Application of sampling methodologies to network traffic characterisation*, Proc. ACM SIGCOMM, Sept. 1993

[COA01] Coates, M.; Nowak, R.; *Network Tomography for Internal Delay Estimation*, IEEE International Conference on Acoustics, Speech, and Signal Processing, May, 2001

[CRO95] Crovella, A.; Bestavros, A.; *Explaining World Wide Web Traffic Self-Similarity*, Technical Report TR-95-015, 1995

[DAL99] Dallas Semiconductor; *DS1553: 64k NV Y2KC Timekeeping RAM Manual*, Technical report, Dallas Semiconductor, 1999

[DEM00] DeMeer, H.; Sasse, H.; *Of packets and people a user centered approach to QoS*, IEEE/IFIP International Workshop on QoS' 2000

[DUF98] Duffield, N.; *Applications of Large Deviations to performance analysis with long-range dependent traffic*, Workshop on stochastic modelling and analysis of communication networks, 1998

[EUN03] Eun, D.Y.; Shroff, N.B.; *A measurement-analytic approach for QoS estimation based on the dominant time scale*, IEEE/ACM Transactions on Networking, vol. 11 pp. 225-235, 2003

[EXP94] *Defination of Optimum Traffic Control Parameters and Results of Trials*; EXPERT Deliverable 15; AC094/EXPERT/WP41/DS/R/P/015/B1

[FEI98] Fei, A.; Pei, G.; Liu, R.; Zhang, L.; *Measurements on delay and hop-count of the Internet*, IEEE Globecom'98 – Internet Mini-Conference, 1998

[FLO93] Floyd, S.; Jacobson, V.; *Random early detection gateways for congestion avoidance*, IEEE/ACM Transactions on Networking, vol. (1), pp. 397-413, Aug. 1993

[FRO99] Frogner, B.; Cannara, A.B.; *Monitoring and Prediction of Network Performance*, Advance Issues of E-Commerce and Web-Based Information Systems, WECWIS, 1999. International Conference on , 8-9 April 1999, Page(s): 122 -129

[GAR00] Garroppo, R.G.; Giordano, S.; Pagano, M.; Procissi, G.; *On the relevance of correlation dependencies in on/off characterization of broadband traffic*, IEEE International Conference on Communications, vol. 2, pp. 811-815, Jun. 2000

[GAR01] Garroppo, R.G.; Giordano, S.; Niccolini, S.; Russo, F.; *A Simulation Analysis of Aggregation Strategies in a WF2Q+ Schedulers Network*, The 2nd IP Telephone Workshop, 2001

[GAR94] Garrett, M; Willnger, W.; *Analysis, Modeling and Generation of Self-Similar VBR Video Traffic*, Proc. SIGCOMM '94, Aug. 1994

[GOL94] Golestani, S.; *A self-clocked fair queueing scheme for broadband applications*, Proc. IEEE INFOCOM'94, pp. 636-646, 1994

[GRA00] Gradshteyn, I.S., Ryzhik, I.M.; *Table of Integrals, Series, and Products*, Academic Press, 2000

[HAN94] Han, W.Y.; Lee, S.J.; Han, C.M.; Kim, S.H.; *Queueing analysis for an ATM multiplexer loaded by CBR and On/Off traffic sources*, Singapore ICCS '94, pp. 760-764, Nov. 1994

[HAO98] Hao, F.; Nikolaidis, I.; Zegura, E.W.; *Efficient Simulation of ATM Network with Accurate End-to-End Delay Statistics, Pro. of the 1998 International Conference on Communications (ICC'98)*, Atlanta, June 7-11, 1998

[HEF86] Heffes, H.; Lucantoni, D.; *A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance*, IEEE JSAC, pp. 856-868 Sept. 1986

[HIL02] Hill, J.; *Assessing the Accuracy of Active Probes for Determing Network Delay, Jitter and Loss*, MSc Thesis in High Performance Computing, The University of Edinburgh, 2002

[HOL00] Holness, F.; *Congestion control mechanism within MPLS*, PhD thesis, University of London, 2000

[HON03] Hong, J.; *Traffic Measurement for High-Speed Internet*, 2003 ICAT Workshop, Seoul, Korea, Apr. 2003

[HSU97] Hsu, H.; *Probability, Random variables, & Random processes*, McGraw Hill, 1997

[HUE98] Huebner, F.; Liu, D.; Fernandex, J.M.; *Queueing performance comparison of traffic models for Internet traffic*, IEEE Globecom 98, vol. 1, pp. 471-476, 1998

[IEC03] International Engineering Consortium; *Voice Quality (VQ) in Converging Telephony and IP Networks*, available from http://www.iec.org/online/tutorials/voice_qual

[JAI02] Jain, M.; Dovrolis, C.; *End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput*, in Proceedings of ACM SIGCOMM '02, Pittsburgh, PA, Aug. 2002

[JAM97] Jamin, S.; Shenker, S.; Danzig, P.; *Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service*, Proc. of the Conference on Computer Communications IEEE INFOCOM'97, Apr. 1997

[JEL95] Jelenkovic, P.R.; Lazar, A.A.; *On the dependence of the queue tail distribution on multiple time scales of ATM multiplexers*, Proc. of the 29th Annual Conference on Information Sciences and Systems, Baltimore, Mar. 22-24, pp. 746-752, 1995

[JIA99] Jiang, W.; Schulzrinne, H.; *QoS Measurement of Internet Real-time Multimedia Service*, Technical Report CUCS-015-99, Columbia University, Dec., 1999

[JIAN99] Jiang, Y; Tham, C.K.; Ko, C.K.; *A QoS Distribution Monitoring Scheme for Performance Management of Multimedia Networks*, Proc. IEEE Globecom'99, pp. 64-68, Brazil, 1999

[JUN02] Juniper Networks; *Juniper enterprise specific Class-Of-Service (COS) MIB*, 2002, available from http://www.juniper.net/techpubs/software/junos/junos57/swconfig57-net-mgmt/html/mib-jnx-cos.txt

[KAT91] Katevenis, M.; Sidiropoulos, S.; Courcoubetis, C.; *Weighted round-robin cell multiplexing in a general purpose ATM switch chip*, IEEE Journal on Selected Areas in Communications, vol. 9, pp. 1265-1279, Oct. 1991

[KES99] Kesidis, G.; *Bandwidth adjustments using on-line packet-level measurements*, in Proc. SPIE Conference on Performance and Control of Network Systems, Boston, Sept. 19-22 1999

[KLI94] Klivanski, S.; Mukherjee, A.; Song, C.; *On Long-Range Dependence in NSFNET Traffic*, Technical Report GIT-CC-94-61, Georgia Tech., 1994

[KOG95] Kogon, S.; Manolakis, D.; *Efficient Generation of Long-Memory Signals Using Lattice Structure*, 1995

[KRA02] Kramer. G; *On generating self-similar traffic using pseudo-Pareto distribution*, available from

http://wwwcsif.cs.ucdavis.edu/~kramer/papers/self_sim.pdf

[KUZ01] Kuzmanovic, A.; Knightly, E.; *Measuring Service in Multi-Class Networks*, Proc. of IEEE INFOCOM'01, Apr. 2001

[LEL94] Leland, W.; *et al.*; *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*, IEEE/ACM Trans. Networking, pp. 1-15, Feb. 1994

[LEU03] Leung,C.M.; Schormans,J.A.; Pitts,J.M.; Woolf,A.; *Accurate decay rate prediction for burst-scale queueing in packet buffers*, Electronic Letter, vol. 39(2), pp. 253-254, Jan. 2003

[LIK95] Likhanov, N.; Tsybakov, B.; Georganas, N.D.; *Analysis of an ATM buffer with self-similar ("fractal") input traffic*, IEEE INFOCOM'95, pp. 985-991, Apr. 1995

[LIU02] Liu, E.; *A hybrid queueing model for fast broadband networking simulation*, PhD thesis, University of London, 2002

[LUC01] Luckie, M.; McGregor, A.; Braun, H.; *Towards improving packet probing techniques*, ACM SIGCOMM Internet Measurement Workshop, 2001

[LUC97] Luca, M.T.; Dernpsey, B.J.; Wrege, D.E.; Weaver, A.C.; *(M,P,S) – an efficient background traffic model for wide-area network simulation*, IEEE GLOBECOM '97, vol. 3, pp. 1572-1576, 1997

[MA00] Ma, A.H.; Schormans, J.A.; Pitts, J.M.; Scharf, E.M.; Pearmain, A.J.; Phillips, C.I.; *Design rules and equivalent capacity for buffering of Pareto source*, Electronic Letters, vol. 36(15), pp. 1274-1275, July 2000

[MA02] Ma, A.H.I.; Schormans, J.A.; *Hybrid technique for analysis of multiplexed power-law traffic in broadband networks*, Electronic Letters, vol. 38(6), pp. 295-297, Mar. 2002

[MA03] Ma, A.; Schormans, J.; Cuthbert, L.; *Aggregation technique for networks with power law traffic and application to accelerated simulation*, IEE Proc. Commun., vol. 150(3), pp. 177-183, June 2003

[MAL97] Malomsoky, Sz.; Vidacs, A. and Saito, H.; *Bandwidth control and its applicability based on queue length monitoring*, PMCCN'97 workshop2, 1997

[MAN69] Mandelbrot, B.; Wallis, J.; *Computer Experiments with Fractional Gaussian Noises*, Water Resources Res., vol. 5, pp. 228-67, Feb., 1969

[MIC97] Michiel, H.; Laevens, K.; *Teletraffic engineering in a broad-band era*, Proceedings of the IEEE, vol. 85, pp. 2007-2003, Dec. 1997

[MIL90] Mills, D.L.; *On the accuracy and stability of clocks synchronized by the network time protocol in the internet system*, ACM Computer Communication Review, 20(1):65-75, Jan., 1990

[MIL92] Mills, D.I.; *Network time protocol (version 3) specification, implementation*, Request for Comments 1305, Internet Engineering Task Force, Mar., 1992

[MOL99] Molnar, S.; Vidacs, A.; *Source characterization in Broadband networks*, COST257 mid-term seminar interim report on source characterizstion, 1999

[NIK92] Nikolaidis, I.; Akyildiz, I.; *Source Characterization and Statistical Multiplexing in ATM networks*, Technical Report, GIT-CC 92-94, Georgia Tech., 1992

[NOR00] Nortel Networks; *Enabling ATM Service Level Agreements*, positioning paper, 2000

[NOR94] Norros, I.; *A Storage Model with Self-Similar Input, Queueing Systems*, vol. 16, pp. 387-396, 1994

[NOR96] Norros, I.; Pruthi, P.; *On the applicability of Gaussian traffic models*, The 13th Nordic Teletraffic Seminar, Norway, 1996

[PAD03] Padmanadhan, V.N.; Lili, Q.; Wang, H.J.; *Server-based inference of internet link lossiness*, IEEE INFOCOM'03, vol. 1, pp. 145-155, 2003

[PAP02] Papagiannaki, K.; Moon, S.; Fraleigh, C.; Thiran, P.; Tobagi, F.; Diot, C.; *Analysi of measured single-hop delay from an operational backbone network*, IEEE INFOCOM, 2002

[PAR01] Park, L.T.; Beak, J.W.; Hong, W.K.; *Management of service level agreements for multimedia Internet service using a utility model*, IEEE Communications Magazine, vol. 39(5), pp. 100-106, May 2001

[PAR93] Parekh, A.; Gallager, R.; *A generalized processor sharing approach to flow control in integrated service networks*, IEEE ACM trans. Networking, vol. 1, pp. 344-357, 1993

[PAX95] Paxson, V.; Floyd, S.; *Wide Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transactions on Networking, June 1995

[PAX96] Paxson, V.; *End-to-end routing behaving in the internet*. SIGCOMM Symposium on Communications Architectures and Protocols, Stanford, California, Aug. 1996

[PAX96a] Paxson, V.; *Towards a Framework for Defining Internet Performance Metrics*, in Proc. INET'96, Montreal, 1996

[PAX98] Paxson, V.; Almes, G.; Mahdavi, J.; *Framework for IP Performance Metrics*, Internet Engineering Task Force, Request for Comments RFC 2330, May, 1998

[PIT00] Pitts, J.M.; Schormans, J.A.; *Introduction to IP and ATM design and performance*, Wiley, 2000

[PIT96] Pitts, J.M.; Schormans, J.A.; *Introduction to ATM design and performance*, Wiley, 1996

[PRE00] Presti, F.L.; Duffield, N.G.; *Multicast-Based Inference of Network-Internal Delay Distributions*, Preprint AT&T Laboratories and University of Massachusetts

[PRU95] Pruthi, P.; Erramilli, A.; *Heavy-tailed ON/OFF source behaviour and self-similar traffic*, IEEE International Conference on Communications, vol.(1), pp. 445-450, Jun. 1995

[PRU95a] Pruthi, P.; *An Application of Chaaotic Maps to Packet Traffic Modeling*, PhD thesis, Royal Institute of Technology, Dept of Teleinformatics, 1995

[RIB00] Ribeiro, V.; Coates, M.; Riedi, R.; Sarvotham, S.; Hendricks, B.; Baraniuk, R.; *Multifractal Cross-Traffic Estimation*, Proceedings ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, Monterey, CA, Sept., 2000

[RIB03] Ribeiro, V.J.; Riedi, R.H.; Richard, G., Baraniuk, G.; Navratil, J.; Cottrell, L.; *PathChirp: Efficient available bandwidth estimation for network paths*, Passive and Active Measurement Workshop PAM2003

[ROB01] Roberts, J.W.; *Traffic theory and the Internet*, IEEE Communications Magazine, vol 9, pp. 94-99, Jan. 2001

[ROB94] Roberts, J.; *Virtual spacing for flexible traffic control*, Int. J. Commun. Syst., vol. 7, pp. 307-318, 1994

[ROU98] Roughan, M.; Veitch, D.; Abry, P.; *On-line estimation of the parameters of long-range dependence*, IEEE Globecom'98, vol. 6, pp. 3716-3721, Nov. 1998

[ROU99] Rougham, M.; Veitch, D.; *Measuring long-range dependence under changing traffic conditions*, IEEE INFOCOM'99, vol. 3, pp. 1513-1521, 1999

[RYO02] Ryoki, N.; Kawahara, K., Ikenaga, T., Oie, Y; *Performance Analysis of Queue Length Distribution of Tandem Routers for QoS Measurement*, 2002 Symposium on Applications and the Internet (SAINT) Workshops, 2002

[SAL01] Salvador, P.; Nogueira, A.; Valadas, P.; *Modeling Local Area Network Traffic with Markovian Traffic Models*, 3rd Conference on Telecommunications, Portugal, 2001

[SCH96] Schormans. J.A.; Pitts, J.M.; Clements, B.R.; Sharf, E.M.; *Approximation M/D/1 for ATM CAC, buffer dimensioning and cell loss performance*, Electronic Letters, vol. 32(3) pp. 164-165 Feb., 1996

[SEU00] Seung, Y.N.; Dan, K.S.; *Measurement-based delay performance estimation in ATM networks*, IEEE GLOBECOM'00, pp. 1766-1770, 2000

[SHA03] Shalunov, S.; Teitelbaum, B; Karp, A.; Boote, J.W.; Zekauskas, M.; *A One-way Delay Measurement Protoco (OWAMP)l*, Internet Draft, May 2003

[SIL98] Siler, M; Walrand, J.; *Monitoring quality of service: measurement and estimation*, Proc. of the 37th IEEE Conference on Decision and Control, vol. 1 pp. 539-544, 1998

[STE02] Stewart, R.A.; *End-to-end delay analysis for small/medium scale IP networks*, PhD thesis, University of London, 2002

[STE03] Stephan E., Jewitt J.; *IPPM reporting MIB*, Network Working Group, Internet Draft, draft-ietf-ippm-reporting-mib-02.txt

[SUB01] Subramanian Mani; *Network Management – Principles and Practice*, Addison Wesley, 2001

[TAQ86] Taqqu, M.; Levy, J.; *Using Renewal Processes to Generate Long-Range Dependence and High Variability*, Dependence in Probability and Statistic, pp. 73-89 Boston, MA, 1986

[TAQ97] Taqqu, M.S.; Willinger, W.; Sherman, R; *Proof of a Fundamental Result in Self-Similar Traffic Modling*, Computer Communication Review, vol. 27, pp. 5-23, 1997

[TEL01] TeleManagement Forum – *SLA Management Handbook*, GB 915, June 2001

[TIM03] Timotijevic,T;Leung,C.M.;Schormans,J.A.; *Accuracy of measurement techniques supporting QoS in packet based Intranet and Extranet VPNs*, to be appeared in IEE proc. Communication Special Issue Superhighways Technology and Broadband VPN vol. 150(4), 2003

[VER99] Verma, D; *Support service level agreement on IP network*, Macmillan Technology Series, 1999

[VLE95] Vleeschauwer, D.; *Experimental verification of the statistical independence of cell delays introduced in consecutive switches*, B-ISDN teletraffic modelling symposium, pp. 105-116, 1995

[WOR02] WorldCOM – *Latency Statistics*; available from http://www.worldcom.com/terms/service_level_guarantee/t_sla_latency.phtml

[YAT93] Yates, D.J.; Kurose; J.F., Towsley, D.; Hluchyj, M.G.; *On Per-session End-to-end Delay Distributions and the Call Admission Problem for Real-time Applications with QOS Requirements*, ACM SIGCOMM Symposium on Commuications Architectures and Protocols, pp 2-12, Sep., 1993

[ZHA00] Zhang, Y.; Paxson, V.; Shenker, S.; *The stationarity of internet path properties: Routing, loss, and throughput*, ACIRI Technical Report, May 2000

[ZSE02] Zseby, T.; *Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurement*, PAM2002